isel Microstep Controller C 142-4.1

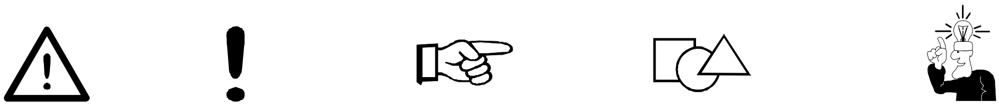
Hardware Manual

B.38331x/99.50/E

On this Manual

Various symbols are used in this Manual to quickly provide you with brief information.

Danger	Caution	Note	Example	Additional Information



Despite all care, printing errors and mistakes cannot be ruled out completely.
Suggestions for improvement and notes on errors are always welcomed.

isel machines and controllers are CE compliant and are marked accordingly.
Any other machine parts and components subject to the CE safety guidelines may not be commissioned unless all relevent standards are fulfilled.

***isel**automation* shall not accept any liability for any modifications on the device by the customer.

The limit values specified in the Certificate of Conformity only apply to the original configuration from works.

Manufacturer: Co. ***isel**automation* KG
In Leibolzgraben 16
D-36132 Eiterfeld

Fax: +49-6672-898-888
E-Mail: automation@isel.com
http://www.isel.com

Contents

1 Introduction 4

2 Safety Notices 5

3 Technical Specifications 6

4 System Description 7

4.1 Block Diagram 7

4.2 Modules and Function Elements 8

4.3 Connectors 9
4.3.1 Serial Interface 9
4.3.2 Motor Output 9
4.3.3 Mains Input 11
4.3.4 Remote Connector 11
4.3.5 PE Conductor / Equipotential Bonding 12
4.3.6 X2 Connector 12
4.3.7 Signal Coupling 13
4.3.8 Step Resolution Settings 14

4.4 Operator Controls 15

5 Start-Up 17

5.1 Application Notes 17

6 Certificate of Conformity 20

***isel*-Interface Card series .... Appendix 1**

***isel*-Stepper Motor Control Card UME 7008 .... Appendix 2**

***isel*-Power Block PB 600-C .... Appendix 3**

***isel*-CNC Operating System 5.x .... Appendix 4**

1 Introduction

The Model C 142-4.1 Stepper Motor Controller is a control unit for three bipolar stepper motors.

In conjunction with a powerful user software, the controller can be used to control three-dimensional motion sequences.

The controller has a processor card, three power output stages and an AC power supply unit with monitoring of safety-relevant components.

The operating system of the processor card (UI 5.C-I/O interface card) can be used for programming the controller both in CNC mode (memory mode) and in DNC mode (direct-style variant). The data can thus either be converted directly or stored in a static RAM.
A battery (optional) is installed to save the RAM data also after a failure of the supply voltage. Moreover, the processor card supports an interchangeable checkcard memory.

The operating system provides, in addition to pure positioning commands, also the processing of eight optically isolated signal inputs and 16 relay switching outputs.
The controller has a serial RS 232 interface for connection with a control computer.

The controller complies with the EMC regulations.

Fig. 1: The Model C 142-4.1 Stepper Motor Controller

2 Safety Notices

The device must be installed and used in accordance with the standards provided in the Certificate of Conformity. The standards and limit values observed by the manufacturer do not protect from improper use of the device.

Therefore, ...

... **you should carry out all connection and installation works on the device only if the device is completely dead, i.e. the device is switched off and the mains supply cable is removed.**

... **all works should exclusively be carried out by expert personnel. Observe, in particular, the regulations and instructions of the electrical industry, as well as the rules for the prevention of accidents.**

Standards for the Stepper Motor Controller used as a basis for the instructions:

EN 60204 (VDE 0113) Part 1 (1992 edition)
- Electrical Equipment of Industrial Machines

EN 50178 (VDE 0160)
- Completion of Electrical Power Installations with Electrical Equipment

VDE 0551
- Regulations for Safety Isolating Transformers

EN 292 Parts 1 and 2
- Safety of Machinery

EN 55011 (VDE 0875)
- Radio and Television Interference Suppression, Limit Value B

IEC 1000-4 (Parts 2-5)
- Inspection and Test Procedures of Noise Immunity

3 Technical Specifications

Housing

- Sheet-steel enclosure with housing consisting of powder-coated aluminium half-shells, W = 475, H = 186, D = 410 mm

isel **Interface Card UI 5.C-I/O**

- 8-bit-micro controller with stepper motor, operating system 5.1
- 3-dimensional linear interpolation and circular interpolation for two of three axes
- Positioning speed max. 10,000 steps/sec.
- 32 KB data memory, battery for data backup as an option
- 8 optically isolated signal inputs and 16 relay switching outputs
- Prepared for use of a 32 KB checkcard memory
- Serial interface to RS 232

isel **Stepper Motor Control Card UME 7008**

- Bipolar power output stage for a 2(4)-phase stepper motor
- Current stabiliser operating at a chopper frequency of 20 kHz
- Phase current max. 8.0 A, short-circuit-proof
- 70 VDC operating voltage

isel **Power Block PB 600-C**

- 650 VA torroidal-core transformer with temperature control and electronic peak making current limiting
- Safety circuit monitoring to EN 292 with EMERGENCY STOP and ON pushbutton input
- VDE Inspection Certificate with manufacturing control (VDE 0160)

DC Power Supply Unit NT 24

- Enclosed built-in power supply unit with torroidal-core transformer
- Output power + 24 V/2.6 A, stabilised

4 System Description

4.1 Block Diagram

For connection with external devices/units, the Stepper Motor Controller is provided with diverse connectors.

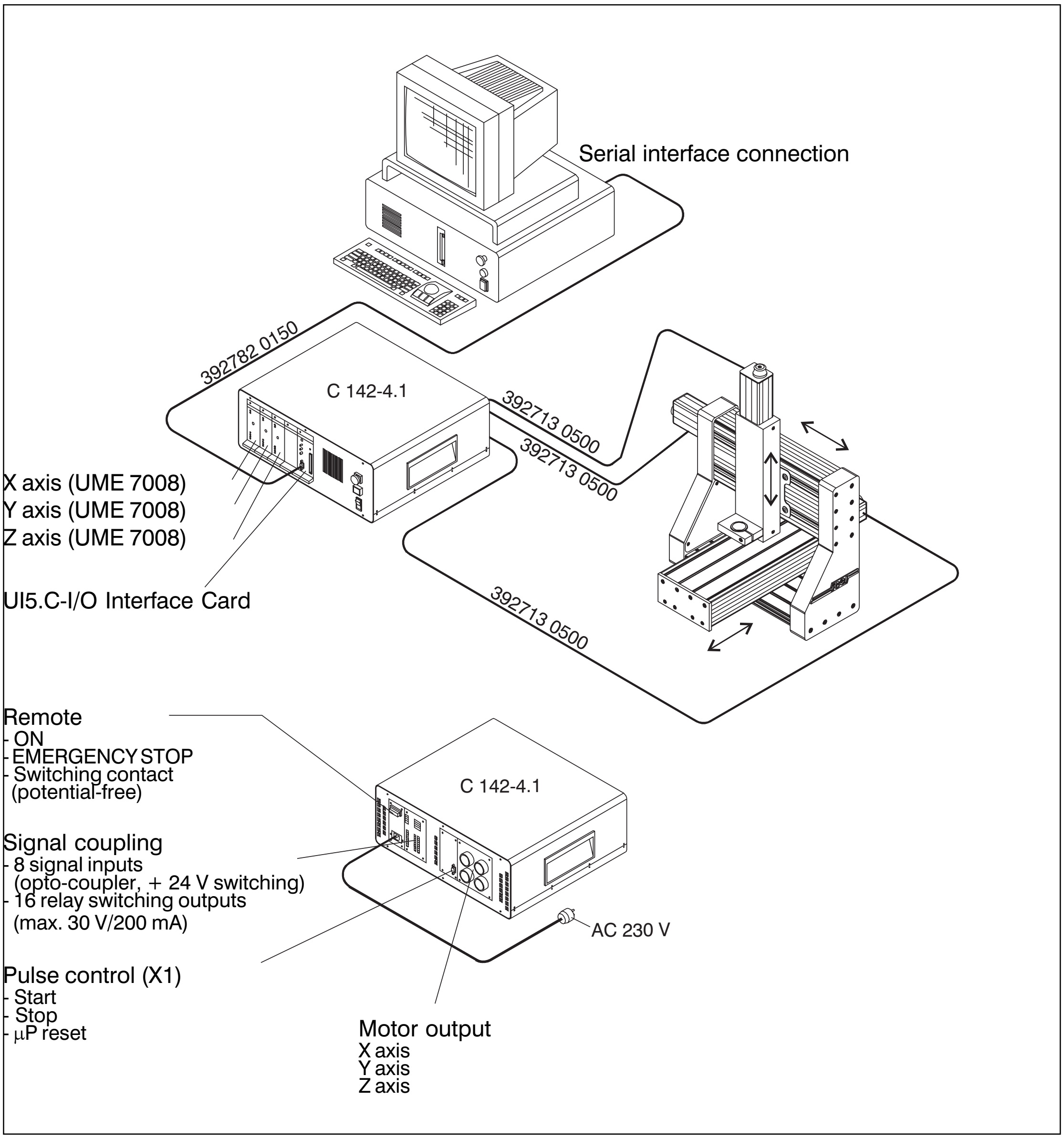


Fig. 2: Connecting the Model C 142-4.1 Stepper Motor Controller

4.2 Modules and Function Elements

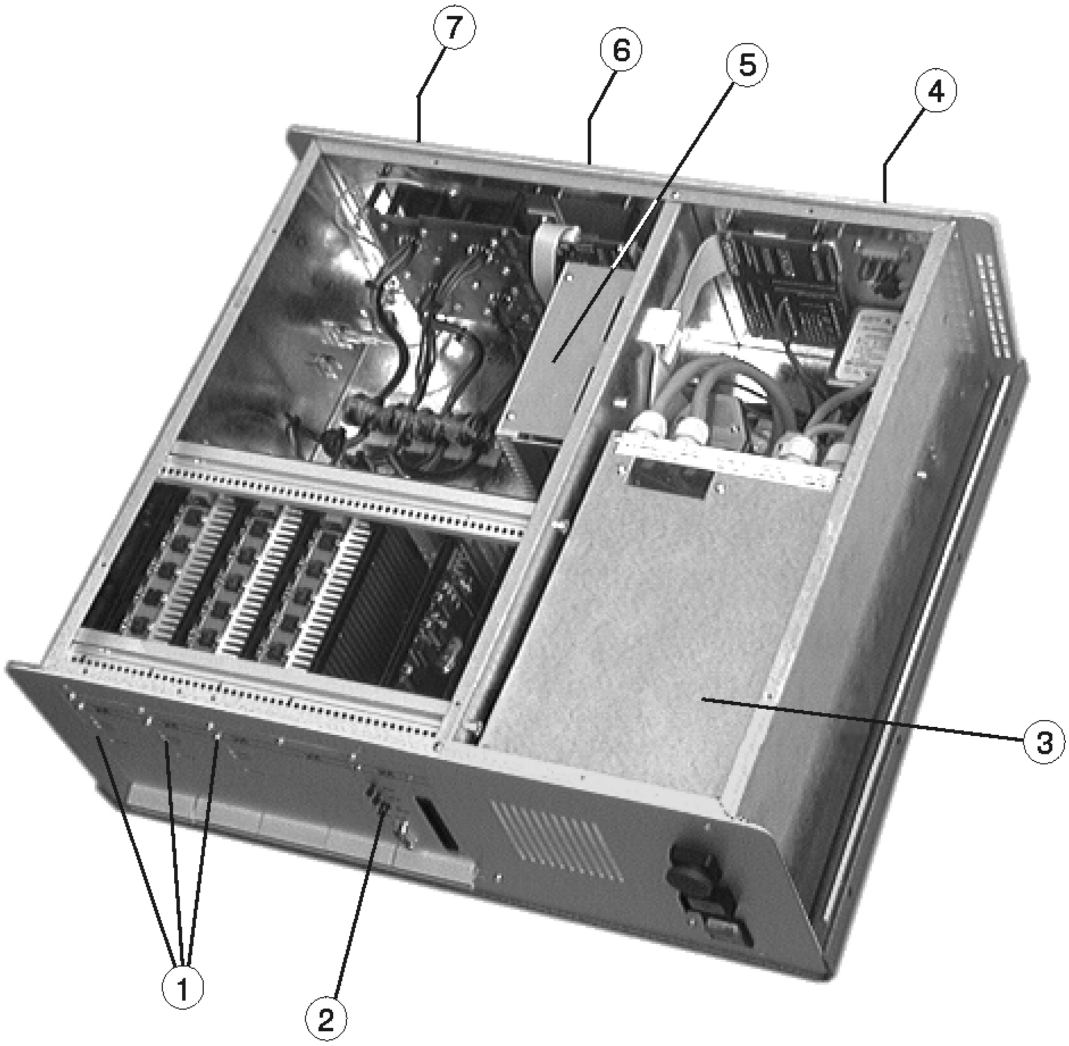


Fig. 3: Model C 142-4.1 Stepper Motor Controller

① Stepper motor power output stage UME 7008

② Interface card UI 5.C-I/O

③ Power block PB 600-C

④ Mains input

⑤ DC power supply unit NT 24

⑥ I/O expansion unit

⑦ Connector to the stepper motors

4.3 Connectors

4.3.1 Serial Interface

The front connector of the interface card serves for connecting to the serial interface of your control computer.

The pin assignment of the 9-pin Sub D male connector is as follows:

Signal	Pin	Pin	Signal
Signal ground (GND)	1	6	not assigned
Receive Data RxD	2	7	not assigned
Transmit Data TxD	3	8	not assigned
not assigned	4	9	not assigned
Logic voltage + 5 V*	5		

* The + 5 V voltage output is intended for the power supply of the optional program selection unit.

4.3.2 Motor Output

Use the circular connectors on the rear of the controller for connecting stepper motor and reference switch.

Pin assignment of the 15-pin circular connector
(Amphenol Tuchel, C16-3 series, housing size 1)

1	O	Motor phase 2B
2	O	Motor phase 2A
3	O	Motor phase 1B
•	O	Motor phase 1A
4	O	Connection for magnetic brake (+ 24 V)
5	O	Auxiliary voltage (+ 24 V)
6	O	Connection for magnetic brake (GND)
7		Functional earthing (cable shield)
8		Not assigned
9	I	Reference switch (n.c. contact, + 24 V)
10		Not assigned
11		Not assigned
12		Not assigned
13		Not assigned
14		Not assigned

O - signal output
I - signal input

You should use shielded cable for the motor connection cables, the braided screen of which is to be connected to the housing potential both on the controller end and on the motor end.

The braided screen of the motor cable, which is connected to both ends, does not constitute a connection to a PE conductor or an equipotential bonding of the units, but is only intended as a functional earthing.

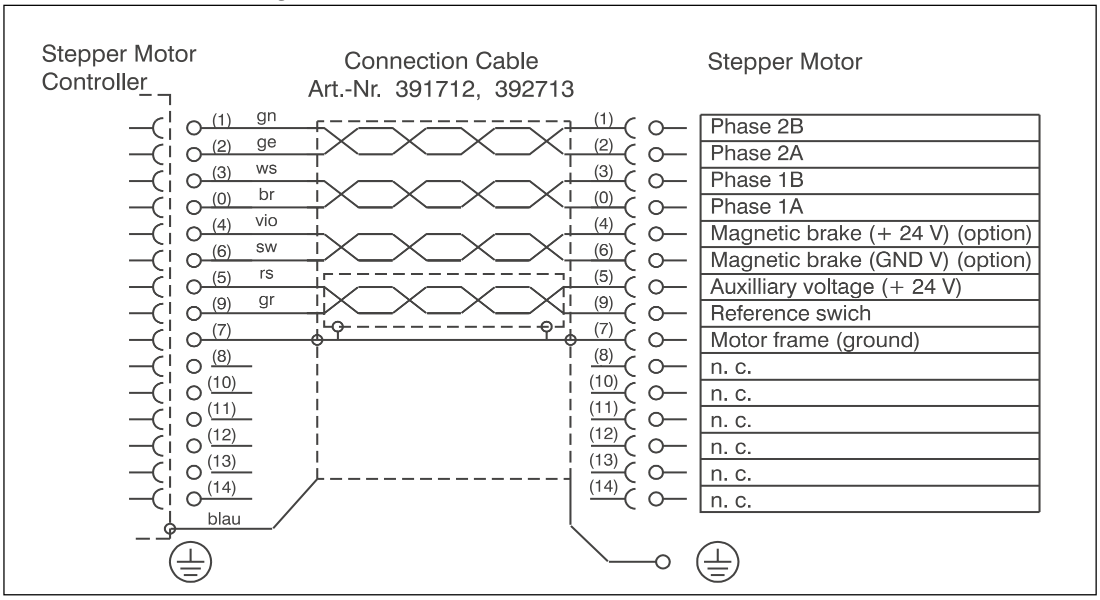


Fig. 4: Design of the motor connection cable

- **Motor phases**
  The outputs 1A and 1B, as well as 2A and 2B are the motor outputs of the controller. They must be connected to the motor phases true to the signals.

- **Reference switch evaluation**
  Reference switches are intended to determine the machine zero. After the reference point approach has been carried out, all positioning instructions in the absolute unit system will refer to this zero point.
  The signal voltage of the switches is + 24 V (plus-switching).

- **Magnetic brake**
  A brake is recommended if the moments of force acting on the drive axis are greater than their holding torques. This can already occur, e.g. with one drive axis installed vertically and the operating voltage of the controller switched off or in case of mains power failure.
  The control voltage of the brake (+ 24 V) is controlled directly from the interface card via a relay. The voltage must be supplied from the rear using the 2-pin plug connector. If necessary you can pick up the voltage on the
  I/O signal coupling.

- **Functional earthing**

  The additional cable brought out from the connector is linked with the braided shield of the cable. This serves for functional earthing of the units and must be connected along the rear side to the threaded bolt (marked with the earthing mark).

  ☞ To avoid misconnections, the coding of the connector in the stepper motor controller is assigned to code 6.

4.3.3 Mains Input

At an operating voltage of 230 V/ 50 Hz, the controller has a total current consumption of approx. 3.0 amperes.

☞ An AC 125 V/60 Hz variant of the controller is also available. In this case, the nominal current consumption is approx. 6.0 A.

4.3.4 Remote Connector

(Phoenix Contact, Mini Combicon (grid 3.81) with cable housing)

The remote connector can be used to connect an external EMERGENCY STOP switch and an OFF switch.

Pin connector assignment:

1 - 2 —— potential-free switching contact (n.o. contact, output)
3 - 4 —— EMERGENCY STOP system (n.c. contact, input)
5 - 6 —— ON button (n.o. contact, input)

- Potential-free switching contact (1 - 2)

  The potential-free switching contact serves to integrate the controller into higher-level EMERGENCY STOP systems. The contact is closed until the power output stages are powered.

- EMERGENCY STOP system (3 - 4)

  The input is intended to connect an external safety device (EMERGENCY STOP switch, safety switch, etc.). If you do not need this input, you should jumper the contact pair.

  ☞ The terminals are supplied with the voltage of the safety circuit. It is imperative to use a potential-free NORMALLY CLOSED contact (n.c. contact) as the switching element. Otherwise, a short circuit may occur in the safety circuit.

- ON button (5 - 6)

  The switching contact is connected to the front-end ON button in parallel and enables the operating voltage provided all safety requirements are fulfilled.

☞ Since acc. to the Machine Protection Regulations only one ON button is permitted in the safety-relevant part of a control system, an external ON button may only be connected if the front-end ON button is disabled by appropriate arrangements (mounting position of the controller, covering of the switch, etc.).

4.3.5 PE Conductor / Equipotential Bonding

For equipotential bonding, the individual function units of a drive system must have a low-impedance connection to the PE conductor.

Acc. to VDE 0113, all chassis parts of the electrical equipment and frame parts of the machine must be linked with the PE conductor.

Furthermore, the equipotential bonding is necessary in order to maintain the limit values specified in the Certificate of Conformity.

4.3.6 X2 Connector

The 9-pin Sub D-female connector can be used to connect external switching elements whose function is similar to those of the processor card.

Pin connector assignment:

Signal		Pin	Pin		Signal
Processor reset	I	1	6	A	+ 24 V
Stop button	I	2	7	A	+ 24 V
Start button	I	3	8	A	+ 24 V
GND	O	4	9	A	GND
+ 24 V	O	5			

- **µP Reset** (contacts 1 - 6)
  Pressing the µP-Reset button initiates a hardware reset of the interface card, thus suddenly terminating all functions of the controller. At the same time, the *Brake* signal output is disabled (+ 24 V control voltage is switched off).
  This function is provided by a pushbutton with a NORMALLY OPEN contact.

- **STOP** (contacts 2 - 7)
  Pressing the STOP button cancels the current program instruction.
  Any stepper motor movement is interrupted by generating a brake ramp.
  This function is provided by a NORMALLY CLOSED contact.

☞ To be able to evaluate the external STOP button, make sure that S 3.5 of the DIP switch on the interface card is set to OFF, as well as S 3.4 and S 3.6 to ON.

☞ If you do not use an external STOP button, you must jumper the contact pair. Otherwise, the controller will change to the STOP condition.

- **START** (contacts 3 - 8)
  The START pulse either carries out a stored instruction set or continues an interrupted instruction sequence.
  This function is provided by a NORMALLY OPEN contact.

The inputs of the X2 connector are optically isolated and operate at a signal voltage of + 24 V.

4.3.7 Signal Coupling

The coupling module serves for connecting external units to the inputs/outputs of the stepper motor controller.

- Signal inputs
  8 optically isolated signal inputs are available.

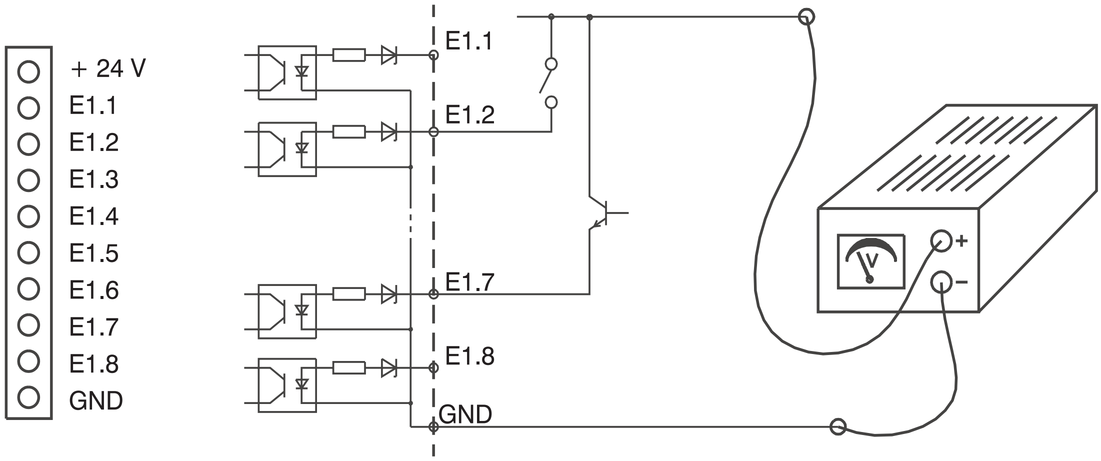


Fig. 5: Components connected to the signal inputs

A 12 V Zener diode and a series resistor are connected to the inputs.
This results in a signal input voltage of + 24 V.
LEDs are provided to indicate which inputs are connected.

The input current of the signal input is + 20 mA (control voltage + 24 V).

- Signal outputs
  To control valves, relays etc., the controller has 16 relay switching outputs.

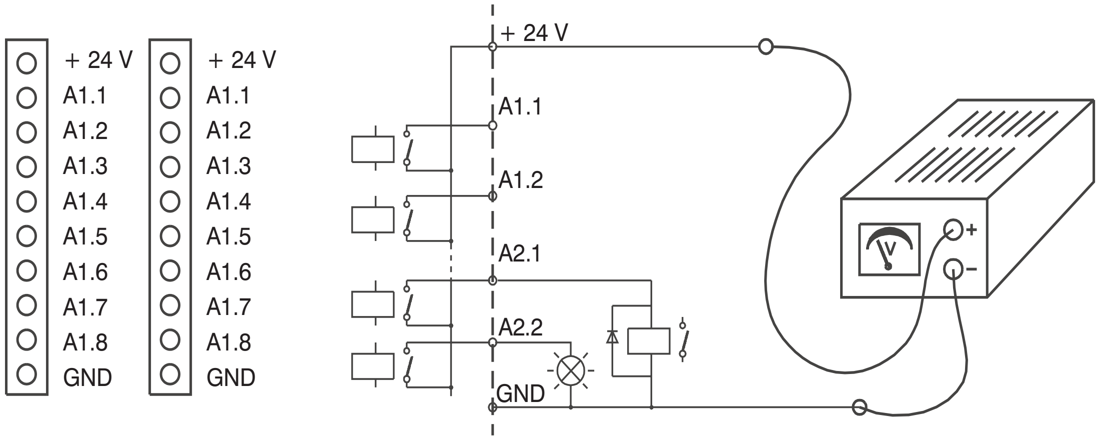


Fig. 6: Signal outputs of the C 142-4.1

The relays used have a maximum load capacity of 50 V at a load current of 200 mA.

The switching contacts of the relays are not included in the entire safety system!

When connecting capacitive or inductive loads, provide for appropriate protective circuits.

Due to the 8-bit memory structure of the interface card, the 16 outputs are divided into 8-bit ports.

For optical control, the signal coupling module has LED bar displays that light if the output is set.

You can pick up the power supply of the signal inputs/outputs (+ 24 V) from the X1 connector (maximum current: 1 A). In case of higher loads, it is absolutely necessary to connect an external power supply unit to the terminals + Vs and GND.

4.3.8 Step Resolution Settings

The default setting on delivery is half-step mode to reduce the resonance properties of the stepper motor system.

4.4 Operator Controls

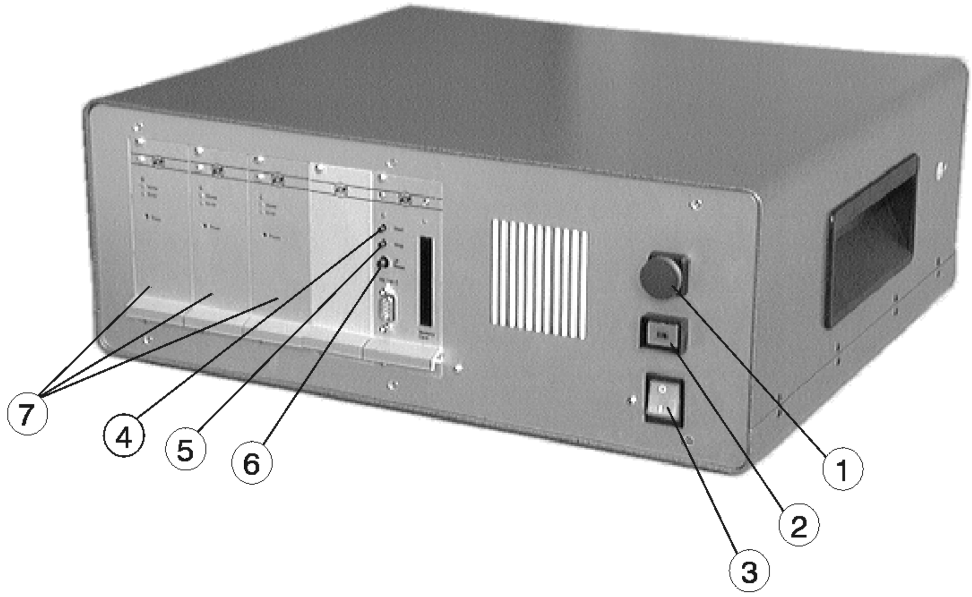


Fig. 8: Front side of the C 142-4.1

① EMERGENCY STOP push-button

② ON button

③ Mains switch

④ Processor reset

⑤ START button

⑥ STOP button

⑦ Phase current potentiometer

① EMERGENCY STOP push-button
The EMERGENCY STOP push-button is a switching element with positive-action contacts. When actuated, this push-button interrupts the safety circuit of the controller, thus switching off the power supply of the power output stages. At the same time, the power transistors of the output stages are disabled and a processor reset of interface card is initiated.

② ON button
With the safety circuit closed, the ON button switches on the power supply of the power output stages. The latching power relay avoids automatic restart of the controller after interrupting the supply voltage.

③ **Mains switch**

If the integrated indicator lights, the stepper motor controller is ready for operation.

④ **Processor reset**

A processor reset will interrupt all activities of the interface card. Any step errors of the stepper motors (due to the abrupt abortion of the step pulse output) will be ignored.
Pressing the µP reset button with simultaneously pressing the START button will initiate a self-test of the controller.

☞ The self-test of the interface card can only aborted by switching off the power supply or pressing µP reset once more.
If the memory card is plugged when a µP reset is carried out, a data field stored there is copied into the static RAM of the processor card.

⑤ **START button**

You can start a CNC data field stored in the data memory by pressing the START button.
A self-test of the controller is initiated in conjunction with the µP reset push-button.

⑥ **STOP button**

Pressing the STOP button interrupts the program sequence of the processor card.
Pressing this button during a positioning movement initiates the brake ramp.
The interrupted process can be restarted by pressing the START button or the @0S command.

⑦ **Phase current potentiometer**

The phase current potentiometer of the power output stage can be used to adapt the output current to the required motor current. The setting range is between 1 A and 8 A or between 1 A and 10 A with current boost activated.

☞ Current boost is the designation of a rise of the motor current during the rotary movement. This will avoid excess heating both of the motor and of the power output stage at a standstill of the motor.

In the C 142-4.1, the appropriate control signal is generated by the interface card.

5 Start-Up

5.1 Application Notes

- After you have turned on the supply voltage of the controller and have then turned on the supply voltage of the output power stages using the ON button, the interface card will remain in the *Reset* state for another 1 ... 2 seconds.
  During this time, you can neither access the process card via the serial interface, nor operate it using the keys. Furthermore, the *Brake* control output is disabled, i.e. a magnetic brake flanged on the motor prevents the motor from rotating. If the START button is pressed within this dead time, a self-test of the interface card is carried out automatically.

- The C 142-4.1 Controller uses the adapted interface card UI 5.C-I/O.
  This is not compatible with the UI 5.0 series. The operating system 5.1, however, remains nearly unchanged so that you can continue using your „old“ programs without restrictions.
  A new feature of this interface card is that the standard software PRO-PAL and PRO-DIN are supported. To make use of this feature, use the supplied software driver i5drv.
  ☞ The software driver i5drv only supports DNC mode.

- The operating system of the interface card can be used to store data of the internal RAM on interchangeable memory media (Memory Card).
  For programming the memory cards, please observe the instructions for the CNC operating system (command @0u).
  ☞ Automatic storage (instruction word: save.) within the data field is not recommended.

- Compared with older stepper motor controllers, the signal voltage of the reference switches has been modified from GND switching to + 24 V switching. The consequence is that the jumper between pin 5 and cable shield in the „old“ cables now results in a short-circuit of the + 24 V supply voltage. In this case, the pin assignment of the connectors must be matched accordingly on both ends (see Section *Motor Output*).

- For setting the stepper motor phase current, the power output stage has a front-end potentiometer.
  The optimum operating current results from the technical specifications of the motor, taking into account the effective power consumption. With a programmed step frequency of approx. 400 Hz in half-step mode, the measuring instrument will display:

  $I_{meas} = I_{phase} \times 0.7 \quad => \quad I_{phase} = I_{meas} / 0.7$

  The default setting of the operating current of the power output stages on delivery is around 4 A.

- In certain operating states, stepper motor drives may tend to resonances resulting either in step losses of individual axes or, in special cases, in a standstill (timeout phases) of the motor what is due to the design and the operating principle of the stepper motor.
  The rotary movement of the stepper motor is carried out by switching the stator field (motor coils) step by step. As a result, the magnetised rotor will accelerate, carry out the step movement, will osciallate to its new position for a short moment and dwell there until the next step pulse is carried out. If the step pulses superimpose the transient characteristics of the rotor, the force vectors will be added.
  The strength and frequency of these resonance signs depend, amongst many other factors, on the mechanical and electrical natural oscillation of the motor, the mechanical design and the link of the two components.
  Since in case of interpolating operation the axis velocities are controlled one against the other, it can not be ruled out completely that at certain vectors system-specific resonances occur. These can be reduced by the following arrangements:
  - Higher acceleration ramps to reduce the dwell time in a resonance range during the acceleration and brake ramp.
  - Use of magnetic or viscous dampers as the basic load (mounted on the drive shaft).
  - Mechanical isolation provided by special couplings using resonance-dampening plastic parts.
  - Use of power-output stages with higher step resolution.
  - Optimisation of the phase current settings.

- The ambient temperature of the controller should not exceed approx. 40 °C.
  Make sure that the vent slots in the bottom plate and in the rear panel are not closed; the resulting heat accumulation would switch off the power output stage.

- The compliance with the EMC limit values requires an equipotential bonding of mechanical and electronic devices with an impedance as low as possible. To achieve this, you should connect both the controller and the numerical axes to a common earthing terminal (cross sectional area of the conductor 2.5 mm^2).

- The supplied motor connection cables of the C 142-4.1 are 5 metres long.
  If you need a different length, you can make it by yourself. When doing so, please pay attention to both the design and the connector pin assignment as per Section 4.3.2. Under no circumstances may the cable length exceed 10 metres.

- The single line brought out from the cable connector is linked with the shielding of the motor connection cable. It serves for functional earthing of the drive unit and not for equipotential bonding. For equipotential bonding, carry an additional, low-impedance connection line from the controller to the numerical drive axis.

- For programming, the interface card has a serial interface to RS 232. A 9-pin Sub-D male connector on the front end is provided as the interface connection.
  To provide a link between interface card and control computer, please use the 3-line shielded cable (for the assignment, see Section 4.3.1). This cable is 1.5 m long and has a Sub-D female connector each on both ends.
  Since the pin assignment of the two connectors is not identical (no 1:1 line), there is the risk to mix up the two connectors. Therefore, they are marked with different colours. Connect the red connector to the control computer, and the gray one to the interface card. In addition, the computer end is marked with an appropriate label.

- The stop button of the pulse control (X2 connector) is only active if the DIP switch S1.5 on the interface card is switched to the OFF position. The switches S1.4 and S1.6 must be set to the ON position.

6 Certificate of Conformity

acc. to the Low Voltage EC Guideline and the relevant EMC Regulations. Doc. No.: k301/95
We, the company

iselautomation KG
Im Leibolzgraben 16
D- 36132 Eiterfeld

declare on our own responsibility that the product

Product designation: **CNC C 142-4.1**
Product No.: **383 310 2003**

to which this Declaration refers complies with the following standard(s) or regulating documents.

1. **EN 50081-1; EN 55011 (VDE 0875)**
   - Electromagnetic Compatibility - Basic Specification on Emitted Interference
   Part 1: Living Area, Business, Trade and Industry, as well as Small Business

   - Limit Values and Testing Methods for Interference Suppression of Scientific and Medical High-Frequency Equipment (Limit Value Class B)

2. **EN 50082-1; IEC 801 (Parts 1-4)**
   - Electromagnetic Compatibility - Basic Specification on Interference Immunity
   Part 1: Living Area, Business, Trade and Industry, as well as Small Business
   - Testing Methods for Interface Immunity

3. **EN 50178 (VDE 0160)**
   Completion of Electrical Power Installations with Electrical Equipment

We herewith assure that the relevant certification procedure has been carried out exclusively in accordance with the Guideline 73/23/EEC (19.02.73), amended 93/86/EEC (22.07.93), in accordance with the Guideline of the Council for the Approximation of the Legal Provisions of the Member States regarding electrical equipment for use within certain voltage limits, in accordance with the Guideline 89/336/EEC (03.05.89), amended 91/263/ECC (29.04.91), amended 2/31/EWG (28.04.92), amended 93/68/EEC (22.07.93) and in accordance with the Guideline of the Council for the Approximation of the Legal Provisions of the Member States on Electromagnetic Compatibility and that the instructions provided in the standard DIN EN 45014 „General Criteria for Certificates of Conformity to be Observed by Providers when Issuing Certificates of Conformity“ have been observed.

isel automation
Hugo Isert · Im Leibolzgraben 16
36132 Eiterfeld · ☎ (06672) 898-0

Eiterfeld, Oct 24, 1995

Rainer Giebel, Electronic Manufacturing Management

Certificate of Conformity

acc. to the Low Voltage EC Guideline and the relevant EMC regulations. Doc. No.: k302/95
We, the company

_isel__automation_ **KG**
Im Leibolzgraben 16
D- 36132 Eiterfeld

declare on our own responsibility that the product

Product designation: **CNC C 142-4.1**
Product No.: **383 311 2003**

to which this Declaration refers complies with the following standard(s) or regulating documents.

1. **EN 50081-1; EN 55011 (VDE 0875)**
   - Electromagnetic Compatibility - Basic Specification on Emitted Interference
   Part 1: Living Area, Business, Trade and Industry, as well as Small Business

   - Limit Values and Testing Methods for Interference Suppression of Scientific and Medical High-Frequency Equipment (Limit Value Class B)

2. **EN 50082-1; IEC 801 (Parts 1-4)**
   - Electromagnetic Compatibility - Basic Specification on Interference Immunity
   Part 1: Living Area, Business, Trade and Industry, as well as Small Business
   - Testing Methods for Interface Immunity

3. **EN 50178 (VDE 0160)**
   Completion of Electrical Power Installations with Electrical Equipment

We herewith assure that the Certification Procedure has been carried out exclusively in accordance with the Guideline 73/23/EEC (19.02.73), amended 93/86/EEC (22.07.93), in accordance with the Guideline of the Council for the Approximation of the Legal Provisions of the Member States regarding electrical equipment for use within certain voltage limits, in accordance with the Guideline 89/336/EEC (03.05.89), amended 91/263/ ECC (29.04.91), amended 2/31/EEC (28.04.92), amended 93/68/EEC (22.07.93) and in accordance with the Guideline of the Council for the Approximation of the Legal Provisions of the Member States on Electromagnetic Compatibility and that the instructions provided in the standard DIN EN 45014 „General Criteria for Certificates of Conformity to be Observed by Providers when Issuing Certificates of Conformity“ have been observed.

isel automation
Hugo Isert · Im Leibolzgraben 16
36132 Eiterfeld · ☎ (06672) 898-0

Eiterfeld, Oct 24, 1995

Rainer Giebel, Electronic Manufacturing Management

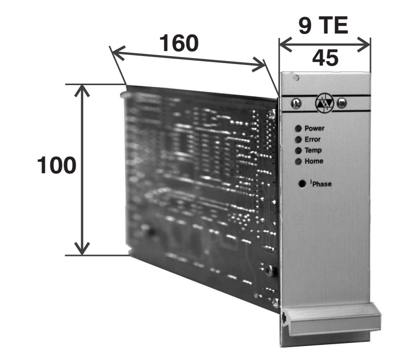
isel Stepper Motor Power Card UME 7008

Introduction

The *isel stepper motor power board* UME 7008 is a micro-step power output stage for bipolar 2(4)-phase stepper motors.
The output stage operates using the bipolar constant current principle and supplies the motor with an adjustable phase current up to 8 A.
A switched-mode power supply operating at approx. 18 kHz provides for low-noise operation and ensures optimum running behaviour of the connected stepper motor.
For controlling, the output stage provides signal inputs for clock, direction, boost and reset. These are designed both as Schmitt trigger inputs (earth reference to supply voltage) and as optically isolated inputs.
The output stages are protected from overtemperature, overcurrent and short-circuit by appropriate protective circuits.
The individual operating conditions are indicated by LEDs on the front panel.
For installation into 19” subracks, the modules are provided with connectors to DIN 41612.

Technical specifications

- Microstep power output stage for a bipolar 2(4) phase stepper motor
- Step resolution switch-changeable, 200, 400, 800, 1600 steps/revolution
- MOSFET output stage
  Short-circuit-proof
  - 8 A continuous current
  - 12 A peak current
- Minimum inductivity 1 mH
- Current setting using a potentiometer on the front panel
- Signal inputs
  - Clock
  - Direction
  - Step resolution
  - Reset
  - Boost
- Optional signal inputs
  - CMOS input with Schmitt trigger, pull-up, low-active
  - 5 V opto-coupler inputs (+ 24 V optional)
- Supply voltage
  + 40 V to + 80 V
- Euro-card 100 x 160 mm with 9 TE front panel
- Connector to DIN 41612
  Series F24/H7
- Signal and pin-compatible with the stepper motor power output stage UMS 6

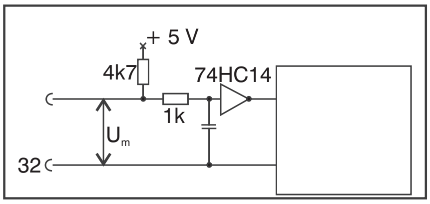


Microstep power output stage UME 7008

Signal description
Inputs

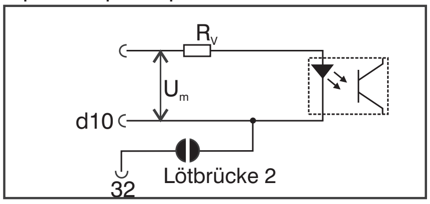
The UME 7008 provides both TTL compabile Schmitt trigger inputs and optically isolated inputs as signal inputs. The signal input stages are defined as follows:

Schmitt trigger inputs:



For controlling, connect the input to 0 V potential (active low)!

Opto-coupler inputs



For controlling, connect the signal input to + 5 V potential and the input GND-Opto to earth (active high).

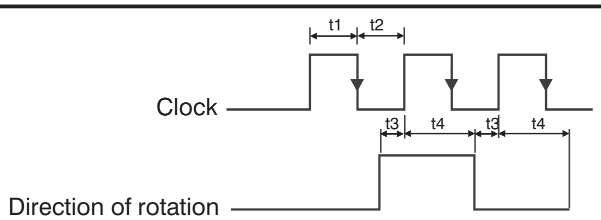
- Upon delivery of the board, soldering jumper 2 is open.
- Upon delivery of the board, the series resistor of the opto-couplers is completed with 330R (signal voltage + 5 V DC).

Technical Specifications

Power supply	+ 40 V DC to + 80 V DC	
Current consumption	typ. 3 A	
Phase current	8 A (continuous current), 12 A (peak current)	
Motor inductivity	min. 1mH	
Signal inputs	CMOS inputs, Schmitt trigger, low-active	
	or alternatively	
	Opto-coupler inputs, + 5 V, high-active	
	- Clock	(Clk/OptoClk)
	- Direction	(Dir/OptoDir)
	- Current boost	(Boost/OptoBoost)
	- Reset	(Reset/OptoReset)
	- Enable	(Enable/OptoEnable)
	- Step resolution 1	(Step1/OptoStep1)
	- Step resolution 2	(Step2/OptoStep2)
Input current		
Opto-coupler	min. 10 mA - max. 25 mA	
Signal outputs	Fault	(Fault)
	Home	(Home)
Controls	Phase current potentiometer	
Display elements	Readiness for operation	(Power)
	Overload	(Error)
	Overtemperature	(Temp)
	Home	(Home)
Protective circuits	Overcurrent, short circuit to earth, short circuit of outputs, overtemperature, overvoltage/undervoltage	
Operating voltage	max. 50 °C	
Shutdown temperature	max. 85 °C	
Dimensions	Euro-card 100 x 160 mm	
Mounting width	9 TE (45.72 mm)	

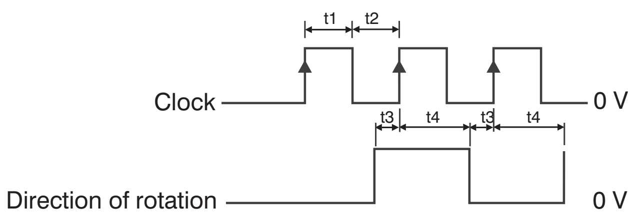
Signal times of the Schmitt trigger

t1 = pulse width	> 5 μs (10 μs at OC)
t2 = interpulse period	> 5 μs
t3 = set-up time direction	> 5 μs (10 μs at OC)
t4 = hold-time direction	> 5 μs (10 μs at OC)
tr = rising edge	< 0.2 μs
tf = falling edge	< 0.2 μs



Signal times of the opto-couplers

t1 = pulse width	> 5 μs
t2 = interpulse period	> 10 μs
t3 = set-up time direction	> 10 μs
t4 = hold time direction	> 10 μs
tr = rising edge	< 0.2 μs
tf = falling edge	< 0.2 μs



Clock (Clk) **z6**
(ClkOpto) **z10**

Every clock pulse with a minimum width of 10 μs results in a defined step angle motion.
The step angle depends on the set resolution and can have the following values:

Full-step mode	1.8 °/pulse
Half-step mode	0.9 °/pulse
1/4-step-mode	0.45 °/pulse
1/8-step mode	0.225 °/pulse

Direction (Dir) **b2**
(DirOpto) **b10**

Signal input for defining the desired direction of rotation of the motor.
H signal - positive direction of rotation of stepper motor (CCW)
L signal - negative direction of rotation of stepper motor (CW)

De-excitation (Ena) **z4**
(EnaOpto) **b12**

An active control signal will disable the stepper motor. The holding torque of the motor will thus be lost; you can turn the motor shaft manually.
The input may only be activated with the motor stopped.

Reset **b6**
(ResetOpto) **d14**

An active control signal will disable the processing of the step pulse and will set the step counter to a defined position (Home position).

Current boost (Boost) **b4**
(BoostOpto) **z12**

An active control signal will raise the motor current and thus the torque in step mode.
If no external protective elements are connected to the input, the current is limited depending on the set phase current.

Step resolution
(Step1, 2) **z2, d4**
(StepOpto1, 2) **b14, d12**

These inputs are used to define the number of steps of a stepper motor per revolution. For a standard 1.8° motor, the following assignment results:

<u>Schmitt trigger inputs</u>

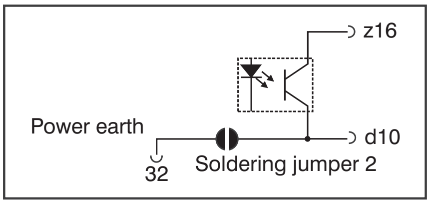
Input Step1	Step2	Number of steps/ Revolution
0 V	5 V	200 (full step)
5 V	5 V	400 (1/2-step)
0 V	0 V	800 (1/4-step)
5 V	0 V	1600 (1/8-step)

<u>Opto-coupler inputs</u>

Input OptoStep1	OptoStep2	Number of steps/ Revolution
5 V	0 V	200 (full step)
0 V	0 V	400 (1/2-step)
5 V	5 V	800 (1/4-step)
0 V	5 V	1600 (1/8-step)

Signal description - outputs

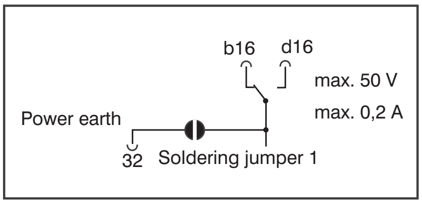
Home **z16**



The opto-coupler output indicates a defined phase position of the stepper motor.
Depending on the step resolution set, the output will close with every
- 4th clock pulse - full step
- 8th clock pulse - half step
- 16th clock pulse - ¼ step
- 32nd clock pulse - 1/8 step

Pin d10 (GND-Opto) is defined as the reference earth.

Fault



The board signals a fault using the Fault relay switch contact. The following error conditions are monitored:
- short-circuit between earth and phase
- short-circuit between the phases
- overtemperature > 85 °C
- undervoltage/overvoltage

If no fault is present, the relay will pick up approx. 1 sec. after the operating voltage has been turned on, closing the contact z14 - d16.

Phase current

The potentiometer I_{phase} on the front panel can be used for linear setting of the phase current.
The control range is between 1.0 A and 8.0 A in normal mode.
For torque compensation in half-step mode, the phase current is raised automatically.

You can measure the phase current using an AC measuring instrument. To do so, connect the instrument in series in one of the stepper motor lines. With a programmed step frequency of approx. 400 Hz in half-step mode, the measuring instrument will display:

$I_{meas} = I_{phase} \times 0.7 => I_P = I_M/0.7$

To determine the phase current using a multimeter, connect the multimeter to a motor phase and measure the phase current at standstill (directly after switching on the unit; the Home LED will light).

Application notices

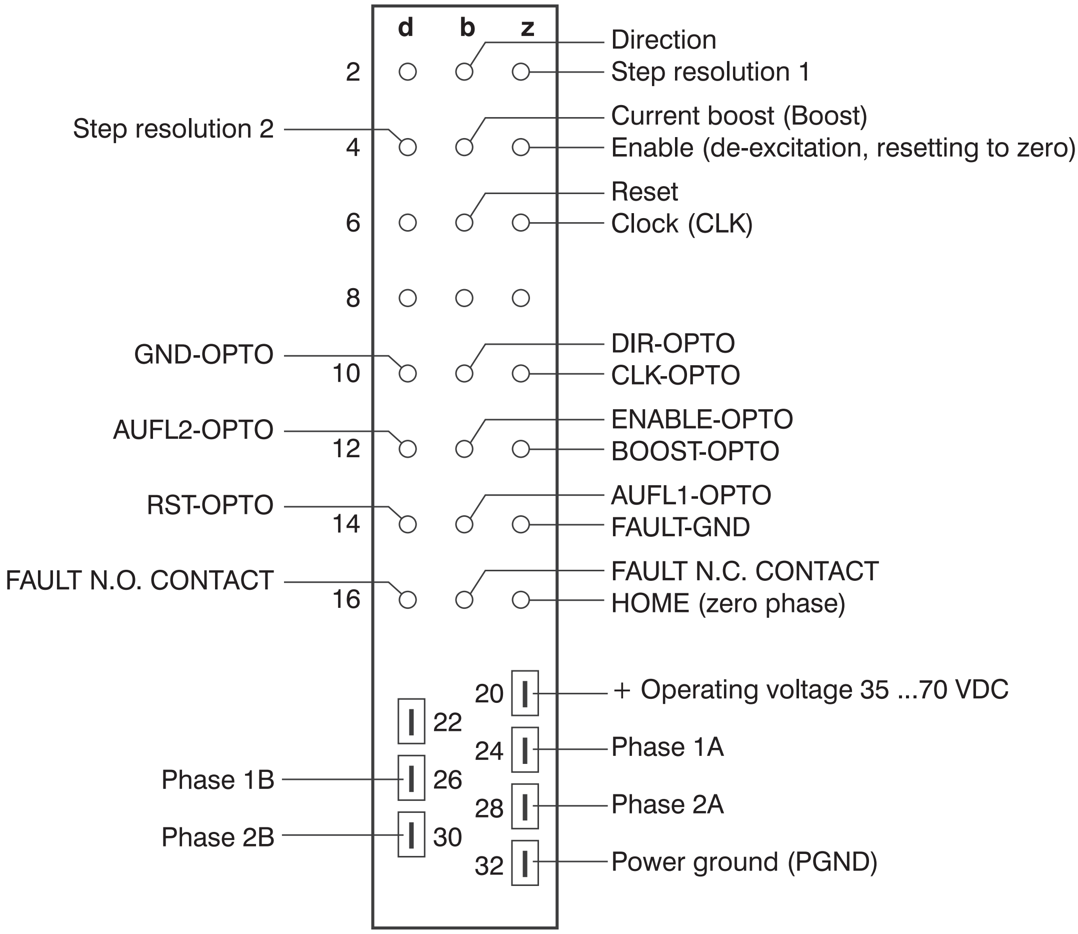
- In case of a fault, the stepper motor output stage is disabled immediately.
  The fault is indicated by the *Error LED* on the front panel and signalled at the fault output. The fault condition remains stored. To reset the fault, turn off the power supply and on again.

- At higher phase currents or higher ambient temperatures, the power output stage must be ventilated externally. To do so, carry an air stream over the cooling face of the board. If the heat sink exceeds a temperature of 85 °C, the output stage is switched off.

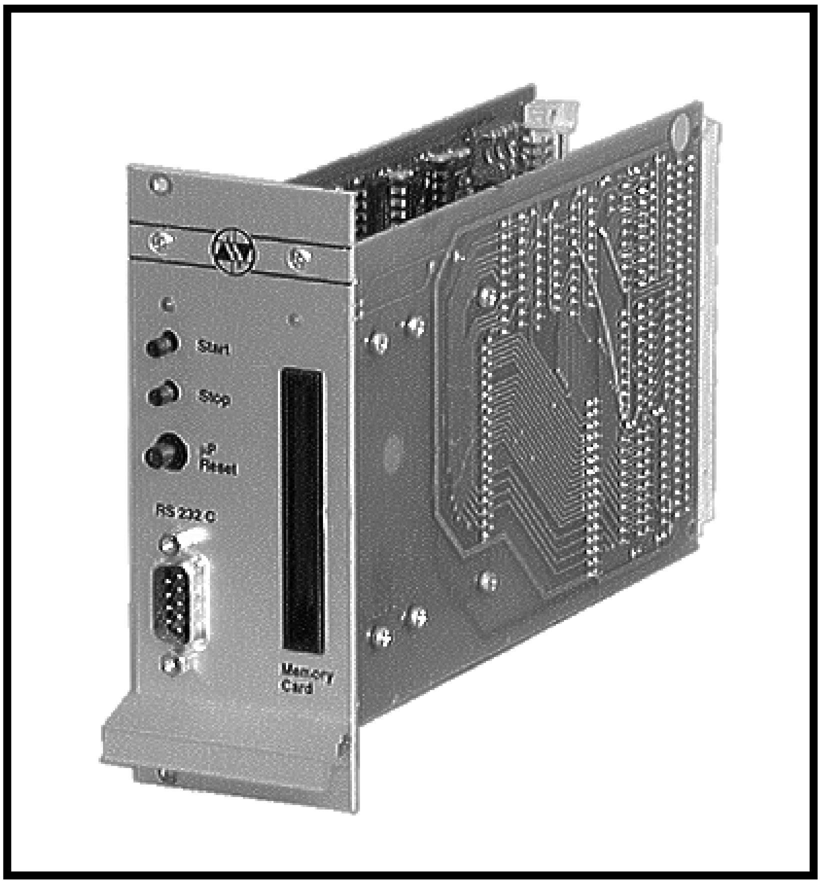
- The signal earth of opto-coupler inputs (Pin d10), Home output (Pin z16) and fault output are potential-free. They can, however, be connected to the power earth by connecting the soldering jumpers BR.1 and BR.2.

- The signal earth of the Schmitt trigger inputs refers to the power earth (Pin z32).

Pin connector assignment - DIN 41612, series F24/H7



isel-Interfacekarten-Serie



Hardware-Beschreibung

B.325xxx.03/2000.12

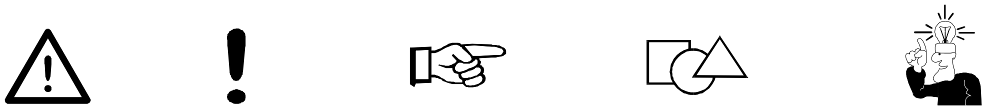
Diese Dokumentation gilt für folgende Baugruppen:

Art.-Nr.:	325 000	-	Interfacekarte UI 4.0
Art.-Nr.:	325 001	-	Interfacekarte UI 4.C
Art.-Nr.:	325 500	-	Interfacekarte UI 4.0-E/A
Art.-Nr.:	325 501	-	Interfacekarte UI 4.C-E/A
Art.-Nr.:	325 050	-	Interfacekarte UI 5.0
Art.-Nr.:	325 051	-	Interfacekarte UI 5.C
Art.-Nr.:	325 550	-	Interfacekarte UI 5.0-E/A
Art.-Nr.:	325 551	-	Interfacekarte UI 5.C-E/A

Unterschiede der Prozessorkarte liegen nur im eingesetzten Betriebssystem und dem Befehlsumfang der Karte sowie der Taktfrequenz des Prozessors. Eine Übersicht der jeweils nutzbaren Befehle ist in der Programmieranlietung ‘CNC-Betriebssystem 5.x’ enthalten.

In dieser Anleitung finden Sie verschiedene Symbole, die Ihnen schnell wichtige Informationen anzeigen.

Gefahr Achtung Hinweis Beispiel Zusatz-Infos



Trotz aller Sorgfalt können Druckfehler und Irrtümer nicht ausgeschlossen werden.
Für Verbesserungsvorschläge und Hinweise auf Fehler sind wir dankbar.

isel-Maschinen und Controller sind CE-konform und entsprechend gekennzeichnet. Für alle sonstigen Maschinenteile und -komponenten, auf die CE-Sicherheitsrichtlinien anzuwenden sind, ist die Inbetriebnahme solange untersagt, bis alle entsprechenden Anforderungen erfüllt sind.

Die Firma ***isel**automation* übernimmt keine Gewähr, sobald Sie irgendwelche Veränderungen an dem Gerät vornehmen.

Die in der Konformitätserklärung aufgeführten Grenzwerte gelten nur für die ab Werk gelieferte Originalkonfiguration.

Hersteller: Fa. ***isel**automation* KG
Im Leibolzgraben 16
D-36132 Eiterfeld

Fax: (06672) 898-888
e-mail: automation@isel.com
http://www.isel.com

Inhaltsverzeichnis

1 Einleitung 4

2 Technische Daten 5

3 Systembeschreibung 6

3.1 Bedienelemente 6

3.2 Serielle Schnittstelle 7

3.3 Funktionselemente 8
3.3.1 Einstellung DIP-Schalter S1 (Baudrate) 9
3.3.2 Einstellung der Beschleunigung 9
3.3.3 Einstellung Voll-/Halbschrittbetrieb (Dip-Schalter S2) (Option) 10
3.3.4 Aktivierung Endlagen-/ Überfahrschalter (Dip-Schalter S3) 10

3.4 Programmier-Modus 11

3.5 Spannungsversorgung 11

3.6 Betriebsstörungen 11

4 Anschluss und Inbetriebnahme 12

4.1 Steckverbinder 13
4.1.1 Signaleingänge 14
4.1.1.1 Referenz-Schalter (Ref.Sw.) 14
4.1.1.2 Überfahrschalter (Stop) 15
4.1.1.3 Start (P1.0) 15
4.1.1.4 µP-Reset 15
4.1.1.5 Signalausgänge 16
4.1.1.6 Betriebsart Voll-/Halbschritt (V/H) 16
4.1.1.7 Taktabschaltung 16
4.1.1.8 Takt 16
4.1.1.9 Richtung 16
4.1.1.10 Stromabsenkung 17
4.1.1.11 Bremse 17
4.1.2 Datenspeicher 17

5 Optionen und Erweiterungen 18

5.1 Aufrüstmöglichkeiten 18

5.2 Optionen 18

6 E/A-Erweiterung 19

6.1 Steckerleiste 20

6.2 Signalankopplung 21
6.2.1 Signaleingänge 22
6.2.2 Signalausgänge 23

6.3 Externer Datenspeicher 24

7 Software-Treiber I5DRV 24

1 Einleitung

isel-Interfacekarten sind Prozessorkarten mit einem ausgereiften CNC-Betriebssystem zur Steuerung von bis zu drei Schrittmotoren. Als Euro-Einschub mit 1" Breite (5 TE) und 3 HE Höhe sind sie in allen 19"-Systemen einsetzbar.

- Die Interfacekarte basiert auf einem 8-Bit-Mikro-Controller-System mit 32 kB Betriebs-EPROM und 32 kB Datenspeicher. Eine umfangreiche, praxisorientierte CNC-Betriebssoftware garantiert die einfache Programmierbarkeit.
- Zur Programmierung von Bewegungsabläufen stehen dabei unter anderem Befehle zur relativen und absoluten Positionierung von bis zu drei Schrittmotoren, Nullpunktfahrt und virtuelle Nullpunkte zur Verfügung. Hierbei wird eine lineare 3D-Interpolation genau so unterstützt wie eine zirkulare Interpolation von zwei aus drei Achsen.
- Die maximal erreichbaren Positionier-Geschwindigkeiten liegen zwischen 30 und 10 000 Schritten/Sekunde. Der Wertebereich beträgt dabei 24 Bit, d. h. eine maximale Wegauflösung von ± 8 000 000 Schritten. Zur Ablaufsteuerung stehen die Befehle schachtelbare Schleifen, erzwungene Verzweigungen, Zeitverzögerungen usw. zur Verfügung.
- Darüber hinaus erleichtern einige Hilfsfunktionen den Umgang mit der umfangreichen Software, so z. B. Einzelschrittausführung (Trace-Mode), Positionsrückmeldungen, Ändern der Gerätenummer und Auslesen von Speicherzellen.
- Durch Direktausführung (DNC-Betrieb) oder Speicherbetrieb (CNC-Betrieb) der Befehle sind sowohl Stand-Alone-Applikationen als auch Anwendungen mit Leitrechnern realisierbar.
- Zur Speicherung von Systemvariablen und CNC-Programmen steht ein 32 kB-Datenspeicher zur Verfügung. Durch Einbau eines optionalen Akku wird eine quasi-permanente Speicherung der CNC-Programme möglich.
- Zur Ansteuerung von Schrittmotorleistungsendstufen erzeugen *isel*-Interfacekarten Signale für Takt, Richtung, Stromabsenkung während Motorstillstand, Takt-Stop und Voll-/Halbschrittumschaltung.
- Die Signalpegel sind TTL-kompatibel (+ 5 V-Logik). Ausgangstreiber ermöglichen den parallelen Betrieb mehrerer Leistungsendstufen. Alle Steuersignale werden an der Kartenrückseite über einen 64-poligen Steckverbinder nach DIN 41612 Bauform C geführt.
- Die Programmierung der Interfacekarte sowie die Kommunikation mit anderen Rechnersystemen ist über eine serielle Schnittstelle mit Software-Handshake und 256 Byte Pufferbereich realisiert. Sie ermöglicht eine zuverlässige 3-Draht-Verbindung zu Steuerrechnern, wobei Baudraten von 2 400 Bd bis 19 200 Bd über DIP-Schalter umschaltbar sind.
- Als Bedienelemente sind in der Frontplatte der Interfacekarten Start-, Stop- sowie Not-Aus-Taster integriert. Die Betriebsbereitschaft wird durch eine LED angezeigt.

2 Technische Daten

Abmessungen	Euro-Karte, 100 x 160 mm, Frontplatte 5 TE (1")	
Spannungsversorgung	+ 5 V, ± 5 %, 300 mA (auf + 6 V bis + 12 V umrüstbar)	
Steckverbinder	DIN 41612 Bauform C, 64-polig a + c	
Eingänge	Rechner-Reset	(aktiv-low)
	Referenz-Schalter	(Schmitt-Trigger)
	Überfahrschalter	(Schmitt-Trigger)
Ausgänge	Takt	(3-State-Output)
	Stromabsenkung	(3-State-Output)
	Richtung	(3-State-Output)
	Taktabschaltung	
	Voll-/Halbschritt	
	Portausgang/-eingang	(P1.0)
Datenübertragung	RS 232 C (9-poliger Sub D-Stiftstecker)	

3 Systembeschreibung

3.1 Bedienelemente

Bild 1: Interfacekarte

Betriebs-LED

... leuchtet bei Betriebsbereitschaft der Prozessorkarte.

Start-Taste

... startet die Ausführung eines im Datenspeicher abgelegten CNC-Datenfeldes.
In Verbindung mit dem µP-Reset-Taster wird ein Selbsttest der Prozessorkarte gestartet.

Stop-Taste

... unterbricht die Ausführung einer programmierten Bewegung durch Einleiten einer Bremsrampe. Der unterbrochene Prozess kann mit der Start-Taste bzw. dem Befehl '@0S' fortgesetzt werden.

Not-Aus (µP-Reset)

... unterbricht, bedingt durch einen Prozessor-Reset, sofort alle Aktivitäten der Interfacekarte. Darüber hinaus werden durch einen parallelen Schaltkontakt die Signalausgänge 'Taktabschaltung' auf 0 V-Potential gelegt. Eventuell auftretende Schrittfehler der über Leistungsendstufen angeschlossenen Motoren werden ignoriert.
Durch Betätigen der µP-Reset-Taste bei gleichzeitig gedrückter Start-Taste wird ein Selbsttest der Interfacekarte eingeleitet.
Bedingt durch die Ausführung der µP-Reset-Taste als Tast-Rast-Schalter ist zum "Lösen" des Reset-Zustandes und zur Freigabe des Taktabschaltungs-Ausgangs eine zweite Betätigung des Tasters notwendig.

3.2 Serielle Schnittstelle

Zur Datenübertagung zwischen der Interfacekarte und einem Steuerrechner wird eine serielle Schnittstelle nach RS 232 eingesetzt. Die Verbindung ist über eine 3-Draht-Leitung realisiert; ein Software-Protokoll ermöglicht die fehlerfreie Übertragung der ASCII- Zeichen. Dabei ist es notwendig, dass sich beide Systeme an das im Folgenden beschriebene Übertragungsprotokoll halten.

- Der angeschlossene Steuerrechner sendet einen Befehl, der mit einem Zeilenende-Zeichen [chr(13)] abgeschlossen ist.
- Die Prozessoreinheit quittiert die Ausführung bzw. Speicherung des Befehles durch das Quittierungs-Signal '0' [chr(48)] oder meldet einen aufgetretenen Fehler mit einem ASCII-Zeichen ungleich '0' (vgl. CNC-Betriebssystem 5.0 Kapitel *Fehlermeldungen der Prozessorkarten*).

Als Datenübertragungsparameter sind auf der Prozessorkarte folgende Werte festgelegt:

9 600 Baud (einstellbar)
8 Daten-Bit
1 Stop-Bit
no Parity

Zur Überprüfung des korrekten Anschlusses bzw. der Funktion der seriellen Schnittstelle verfügt die Prozessorkarte über eine Selbsttestroutine. Sie wird ausgeführt, wenn Sie die Start-Taste festhalten und die μP-Reset-Taste kurz betätigen.
Die Interfacekarte überprüft daraufhin ihren Speicherbereich sowie die Schalterstellung des 4-fach-DIP-Schalters. Anschließend werden zum Test des angeschlossenen Schrittmotors einige Taktimpulse ausgegeben. Abgeschlossen wird die Testroutine durch einen permanent gesendeten ASCII-Zeichensatz an der seriellen Schnittstelle.
Durch Betätigen irgendeiner Taste der Rechnertastatur wird dieser Modus abgebrochen und jedes weiterhin von der Prozessorkarte empfangene Zeichen als Echo zurückgesendet.

Der Selbsttestroutine wird durch einen μP-Reset beendet!

Zur Inbetriebnahme der seriellen Verbindung von Steuerrechner und Interfacekarte kann folgendes Basic-Schnittstellen-Testprogramm verwendet werden.

Schnittstellen-Testprogramm z. B. in GW-Basic:

```
100     open"com1:9600,N,1,RS,CS,DS,CD"  as#1
110     if loc(1)0 then print input$ (loc(1),1):
120     a$=inkey$: if a$"" then print #1,a$;:print a$;
130     goto 110
```

Die Pin-Belegung der Steckverbinder

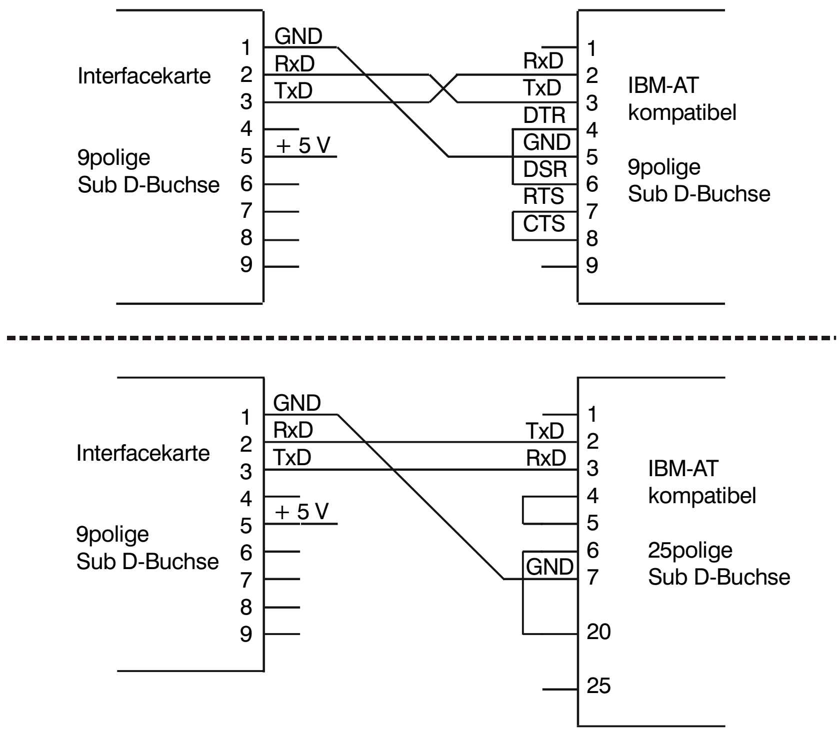


Bild 2: Anschluss serielle Schnittstelle

3.3 Funktionselemente

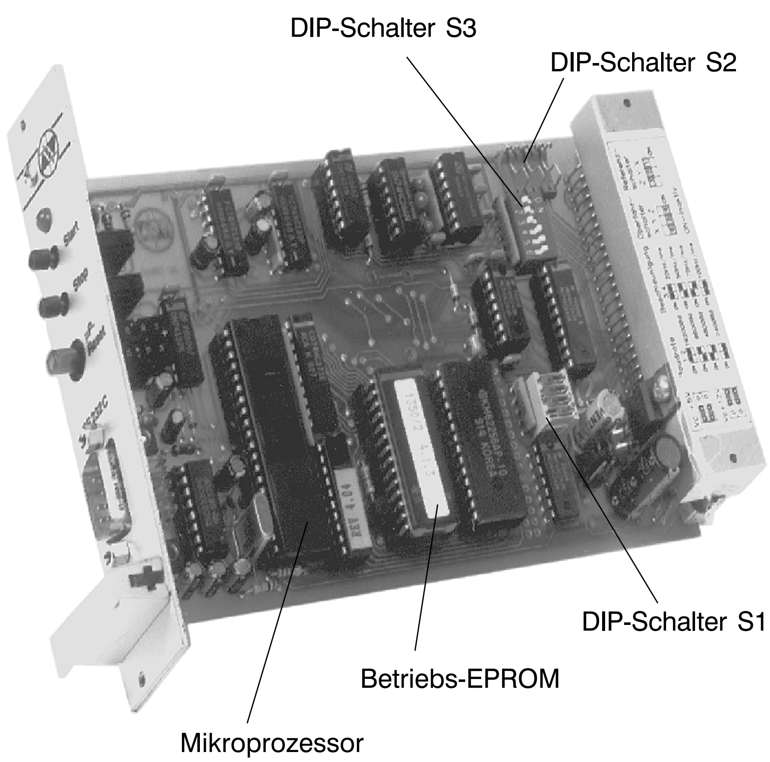


Bild 3: Interfacekarte (ohne E/A-Erweiterung)

3.3.1 Einstellung DIP-Schalter S1 (Baudrate)

Zur Festlegung der Übertragungsrate der seriellen Schnittstelle wird nach jedem Mikroprozessor-Reset die Schalterstellung des 4-poligen Schiebeschalters S1 abgefragt. Dabei ergeben sich aus den vier möglichen Schalterkonfigurationen von Schalter 1 und 2 die unterschiedlichen Baudraten.

S1.1	S1.2	Baudrate
OFF	OFF	2 400 Bd
ON	OFF	4 800 Bd
OFF	ON	9 600 Bd*
ON	ON	19 200 Bd

*** Auslieferungszustand 9 600 Bd**

3.3.2 Einstellung der Beschleunigung

Bei Betrieb eines Schrittmotors außerhalb des Anlaufbereiches ist eine Beschleunigungs- und Bremsrampe erforderlich. Während bei der Beschleunigungsrampe die Schrittfolgefrequenz des Motors kontinuierlich von der Startfrequenz auf die Betriebsfrequenz gesteigert wird, erfordert die Verzögerungsrampe den umgekehrten Vorgang.
Durch unterschiedliche Steigungen lassen sich die Kurven in Bezug auf Beschleunigungszeit und Last optimieren.
Es stehen Ihnen standardmäßig vier verschiedene Rampen zur Verfügung.
Mit Schalter 3 und 4 des 4-poligen DIP-Schalters S1 können Sie die Rampen definieren.

S1.3	S1.4	Rampe
ON	ON	25 Hz/ms
OFF	ON	50 Hz/ms
ON	OFF	75 Hz/ms
OFF	OFF	100 Hz/ms

*** Auslieferungszustand 25 Hz/ms**

3.3.3 Einstellung Voll-/Halbschrittbetrieb (Dip-Schalter S2) (Option)

Dieser Schalter ermöglicht die zentrale Einstellung der Betriebsart der angeschlossenen Leistungsendstufen.

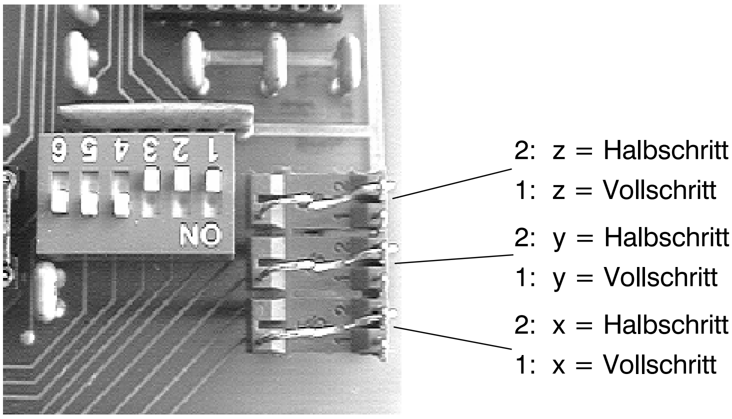


Bild 4: Interfacekarte (Platinenauszug Schalter S2)

Der Schalter S2 wird bei Einsatz der Karte in den Schrittmotor-Controller C 116-4 und C 142-4 nicht ausgewertet. Die Festlegung der Betriebsart wird dort direkt auf der Verbindungsplatine mit Jumper-Steckern vorgenommen.

3.3.4 Aktivierung Endlagen-/ Überfahrschalter (Dip-Schalter S3)

Zur Überwachung von Endlagen- und Überfahrschalter der Schrittmotorantriebs-einheiten werden die Signale der entsprechenden Achsen getrennt auf die Prozessorkarte geführt und dort verarbeitet. Zur Freigabe des Signaleinganges dient der 6-fach-DIP-Schalter S3. Jeder extern zu überwachende Schalter muss durch Umschalten auf OFF aktiviert werden, dementsprechend jeder nicht vorhandene Schalter durch Umschalten auf ON gesperrt werden. Dabei ergibt sich folgende Zuordnung:

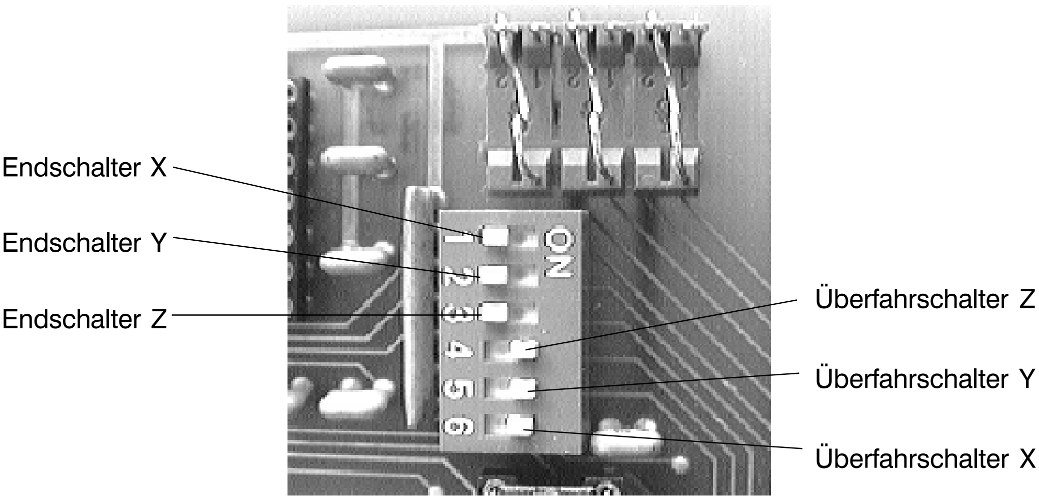


Bild 5: Interfacekarte (Platinenauszug Schalter S3)

3.4 Programmier-Modus

Für einen optimalen Einsatz ermöglicht das Betriebssystem sowohl eine Programmierung im DNC-Modus (direkte Ausführung der übergebenen Befehle) als auch im CNC-Modus (auszuführendes Programm wird im internen Datenspeicher abgelegt und später durch ein Start-Signal gestartet, vgl. CNC-Betriebssystem 5.0).

Im DNC-Modus werden dem Prozessormodul die Bearbeitungsparameter einzeln übergeben und von ihm direkt ausgeführt. Durch Auswertung der Quittierungssignale der IT 108 ist der übergeordnete Steuerrechner in der Lage, kontinuierlich und ohne Begrenzung Daten zu übergeben.

Im CNC-Modus (Speicherbetrieb) wird der Prozessoreinheit ein komplettes Datenfeld übergeben. Die Daten werden nach Erhalt vom Prozessor quittiert und in einem Datenspeicher abgelegt. Die Ausführung des Datenfeldes (ca. 1 800 Befehlssätze) erfolgt anschließend durch Betätigen der Start-Taste bzw. eines Startbefehles des Steuerrechners.

3.5 Spannungsversorgung

Als Spannungsversorgung benötigt die Interfacekarte eine Gleichspannung von + 5 V bei einem mittleren Stromverbrauch von ca. 300 mA. Sie wird über die Steckkontakte a,c30 (+ Vc) und a,c32 (GND) des rückwärtigen Steckverbinders auf die Karte geführt.

Zur Überwachung der Speisespannung befindet sich auf den Prozessorkarte (ab Version 1350/4) eine entsprechende Schaltung, die bei Unterschreiten einer Schwellenspannung den Prozessor zurücksetzt. Dies wird durch gleichzeitiges Verlöschen der Betriebs-LED angezeigt.
Ein DC/DC-Wandler auf der Interfacekarte ermöglicht die Spannungsversorgung mit + 6 V bis + 12 V. Das Umschalten des Eingangsspannung-Levels geschieht durch zwei Jumper (siehe Aufkleber auf dem Steckverbinder der Interfacekarte).

3.6 Betriebsstörungen

Zur Erkennung von Betriebsstörungen verfügt die Interfacekarte hardwaremäßig über einen Unterspannungsdetektor sowie softwaremäßig über Überwachungsmodule für End- und Überfahrschalter sowie über Kommunikations- und Speicherfehler.
Während bei Spannungsfehlern der Mikroprozessor in den Reset-Zustand geschaltet und die Kommunikation zum übergeordneten Rechner abgebrochen wird, erfasst der Prozessor alle anderen Betriebszustände durch das Betriebssystem. Hier erfolgt die Fehleranzeige über die serielle Schnittstelle (Fehlercode vgl. CNC-Betriebssystem 5.0 Kapitel 4, sowie serielle Schnittstelle S. A3).

Fehlercode	Fehlerart	Fehlerbeseitigung
Betriebs-LED leuchtet nicht	- keine Versorgungsspannung 7 angelegt - Versorgungsspannung 4,65 - μP-Reset-Eingang (c28) ist aktiv low	- Versorgungsspannung + 5 V/300 mA an Pin 30 (+ 5 V) und Pin 32 (GND) anlegen - Signaleingang μP-Reset überprüfen
LED in μP-Reset-Taste leuchtet	- Tast-Schalter ist nach μP- Reset eingerastet	- durch nochmaliges Betätigen Tast-Schalter lösen
Karte antwortet nicht	- Verbindungsleitung der RS 232 nicht korrekt gesteckt. - Serielle Schnittstelle der Interfacekarte defekt - Serielle Schnittstelle des Steuerrechners defekt	- Steckverbinder mit dem Aufkleber 'AT-Seite' mit der seriellen Schnittstelle des PC verbinden. - Schnittstellen-Testprogramm (s. S.6) starten und Selbsttest ausführen. - ggf. seriellen Schnittstellen-Baustein (MAX 232) ersetzen. - Überprüfen der Schnittstelle durch Ankopplung eines anderen Gerätes

4 Anschluss und Inbetriebnahme

Zum Einsatz in 19"-Baugruppenträgern (nach DIN 41494) verfügt die Interfacekarten-Serie über einen 64-poligen Steckverbinder DIN 41612 C. Über ihn werden zum einen alle Signaleingänge der Prozessorkarte zugeführt (z. B. Start-, Stop-, Referenz-Schalter), zum anderen von der Prozessorkarte alle Steuerausgänge zur Verfügung gestellt (z. B. Takt und Richtung).
Bedingt durch die Konzeption als Interpolator für max. drei Schrittmotorantriebe sind auf der Prozessorkarte die entsprechenden Signalein- und -ausgänge für jede Antriebsachse getrennt ausgeführt.

4.1 Steckverbinder

Zur Adaption in 19"-Systemgehäusen verfügt die Interfacekarte über eine 64-polige Stiftleiste nach DIN 41612 Bauform C.

Reihe A			Reihe C		
Signal		Pin	Pin		Signal
NC		1	1		NC
NC		2	2		NC
NC		3	3		NC
NC		4	4		NC
NC		5	5		NC
NC		6	6		NC
V/H X-Achse	A	7	7		NC
V/H Z-Achse	A	8	8	A	V/H Y-Achse
Ref.Sw. Y-Achse	E	9	9	E	Ref.Sw. X-Achse
NC		10	10	E	Ref.Sw. Z-Achse
Taktabschaltung X	A	11	11	A	Taktabschaltung Y
Taktabschaltung Z	A	12	12		NC
NC		13	13		NC
+ 5 V**		14	14		Bremse**
RxD*		15	15		TxD*
NC		16	16	A	Richtung X-Achse
Takt X-Achse	A	17	17	A	Richtung Z-Achse
Takt Z-Achse	A	18	18	A	Richtung Y-Achse
Takt Y-Achse	A	19	19		NC
Stromabsenkung Z	A	20	20	A	Stromabsenkung Y
Stromabsenkung X	A	21	21	A	NC
NC		22	22		NC
NC		23	23		NC
NC		24	24		NC
Stop Z-Achse	E	25	25	E	Stop Y-Achse
Stop X-Achse	E	26	26	E	P1.0
P1.0	E	27	27	E	P1.0
NC		28	28		μP-Reset
NC		29	29		NC
+ 5 V		30	30		+ 5 V
NC		31	31		NC
GND		32	32		GND

NC= nicht belegt
A = Signalausgang
E = Signaleingang
* ab Version AZ1350/3
** ab Version AZ1350/4

4.1.1 Signaleingänge

Als Signaleingänge verarbeitet die Interfacekarte folgende Eingänge:

- Referenz-Schalter (Ref.Sw.)
- Überfahrschalter (Stop)
- Start (Start)
- - μP-Reset

4.1.1.1 Referenz-Schalter (Ref.Sw.)

Zur Positionsbestimmung innerhalb eines Schrittmotor-Antriebssystems besteht die Notwendigkeit eines Maschinennullpunktes bzw. Referenzpunktes.
Zur Auswertung von entsprechenden Sensoren verfügt die Interfacekarte über den Eingang *Referenz-Schalter (Ref.Sw.)*. Bei dem Eingang handelt es sich um einen aktiv-high-Eingang, der intern über einen Pull-up-Widerstand auf + 5 V gelegt ist. Die Auswertung des Signales erfolgt, wenn auf dem im Ruhezustand GND-Potential führenden Eingang ein + 5 V-Signal auftritt.

In *isel*-Lineareinheiten hat sich als Referenz-/Endlagenschalter ein Mikro-Schalter (Öffner-Schaltkontakt) durchgesetzt, der zwischen GND und Signaleingang *Ref.Sw.* geschaltet ist.

Wird während einer Verfahrbewegung der Referenzschalter betätigt, stoppt die Prozessoreinheit abrupt die Schrittimpulsausgabe. Erfolgt eine Aktivierung des Schalters während der Ausführung einer Referenzfahrt, wird die Impulsausgabe ebenfalls unterbrochen, jedoch nach Ändern des Richtungsbits mit einer kleinen Schrittfrequenz wieder gestartet.
Ein erneuter Interrupt (durch Verlassen des Schalterbereiches) stoppt den Schrittmotor exakt am Maschinen-Nullpunkt. Hierbei wird eine Wiederholgenauigkeit von ± 1 Schritt erreicht. Bei Verwendung eines induktiven, kapazitiven oder optischen Näherungsschalters ist der Minus-Pol des Sensors mit dem GND-Signal der Antriebseinheit sowie der Signalausgang des Sensors (open-collector) mit dem Steuerungseingang *Ref.Sw.* zu verbinden.

☞
- als Sensor muss ein NPN-Typ eingesetzt werden
- der Sensor muss als Öffner arbeiten (Ruhezustand Ausgang leitend)

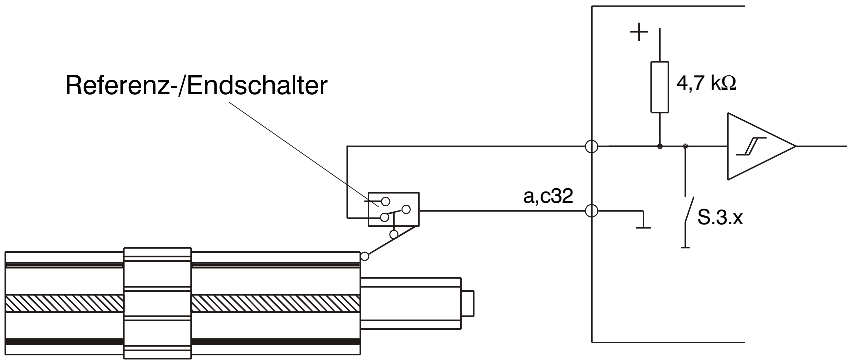


Bild 6: Anschluss Referenzschalter

Bei nicht oder nicht korrekt angeschlossenem Referenz-Schalter meldet die Interfacekarte über die serielle Schnittstelle Fehler '2'.

Bedingt durch die begrenzte Anzahl von Hardware-Interrupts werden auf der Interfacekarte die Signalquellen der drei Referenzschalter-Eingänge miteinander verknüpft. Hierzu sind die Signaleingänge an eine Impulsformungsstufe geführt, die aus jeder Flankenänderung eines Eingangssignales einen definierten Impuls mit 10 μs Impulsbreite erzeugt.
Werden einzelne Referenzschalter nicht benötigt bzw. angeschlossen, ist der entsprechende Signaleingang direkt auf GND-Potential zu legen oder - wie in Absatz 3.4.4 beschrieben, mit Hilfe des DIP-Schalter S3 zu sperren.

4.1.1.2 Überfahrschalter (Stop)

Dieser Eingang führt, genauso wie bei Betätigung des frontseitigen Stop-Tasters, zu einem Stop-Interrupt des CNC-Betriebssystemes. So veranlasst ein negativer Impuls (H-L-Signal-wechsel) am Signaleingang einem gebremsten Abbruch einer Verfahrroutine.

Einsatzmöglichkeiten dieses Einganges sind z. B. in Verbindung mit Referenzschaltern geringer Schalthysterese zu sehen (mechanische Zerstörung durch Nachlaufweg des Schrittmotors bei abrupten Reset mit hoher Geschwindigkeit). Ähnlich dem Signaleingang *Ref.Sw.* werden auch die Überfahrschalter-Eingänge zu einem Interrupt zusammengefasst, sodass die Aktivierung eines Einganges den Bewegungsablauf aller aktiven Schrittmotor-achsen unterbricht.
Zu beachten ist hierbei, dass ein solchermaßen unterbrochener Bewegungsablauf mit der Start-Taste reaktiviert werden kann und ein kontinuierlich offener Signaleingang einen erneuten Interrupt verhindert. Sie sollten deshalb darauf achten, dass ein Überfahrschalter-Eingang nur durch einen kurzen negativen Impuls beschaltet wird.

Analog zum *Ref.Sw.*-Eingang sind auch beim Überfahrschalter-Eingang einzelne, nicht benötigt Signaleingänge direkt auf GND-Potential zu legen oder, wie in Absatz 3.4.4 beschrieben, mit Hilfe des DIP-Schalter S3 zu sperren.

4.1.1.3 Start (P1.0)

Der Signaleingang arbeitet parallel zur frontseitigen Start-Taste. Durch kurzzeitiges Verbinden mit dem GND-Potential wird ein in der Steuerung gespeichertes Programm gestartet.

4.1.1.4 µP-Reset

Der Steuerungseingang μP-Reset liegt schaltungstechnisch parallel zum frontseitigen μP-Reset-Tast-Rast-Schalter. Durch Verbinden des Eingangs mit GND-Potential wird der Mikroprozessor gesperrt und somit alle Aktivitäten unterbrochen. Hierbei werden Positioniervorgänge der angeschlossenen Schrittmotoren abrupt beendet.

4.1.1.5 Signalausgänge

Zur Ansteuerung von Schrittmotor-Leistungsendstufen stellt die Interfacekarte zur Verfügung:

- Betriebsart Voll-/Halbschritt (V/H)
- Taktabschaltung
- Takt
- Richtung
- Stromabsenkung
- Bremse

4.1.1.6 Betriebsart Voll-/Halbschritt (V/H)

Je nach Schalterstellung des 3-poligen DIP-Fix-Schalters liegt an den entsprechenden Signalausgängen entweder + 5 V- oder 0 V-Potential.

Schalterstellung 1 (0 V)	- Vollschrittbetrieb
Schalterstellung 2 (+ 5 V)	- Halbschrittbetrieb

Zur Zuordnung der jeweiligen Schalter siehe Kapitel 3.4.3.

4.1.1.7 Taktabschaltung

Der Signalausgang stellt eine zusätzliche Sicherheit bei einem Hardware-Reset der Interfacekarte dar. Durch Betätigen der frontseitigen μP-Reset-Taste werden neben dem Reset-Impuls für den Mikro-Controller die drei Signalausgänge auf 0 V-Potential geschaltet. In *isel*-CNC-Controllern ist dieser Ausgang auf den jeweiligen Takt-Stop bzw. Reset-Eingang der Schrittmotor-Leistungsendstufe gelegt und bewirkt ein zusätzliches Sperren der Taktverarbeitung.

4.1.1.8 Takt

Am Taktausgang der Interfacekarte stehen - entsprechend des im Mikro-Controller berechneten Frequenzverlaufes der einzelnen Schrittmotoren - die jeweiligen Takte für die Leistungsendstufen zur Verfügung. Als Taktimpuls ist ein positiver Impuls von ca. 10 μs Breite definiert.

4.1.1.9 Richtung

Der Richtungsausgang gibt je nach vorgegebener Drehrichtung des Schrittmotors ein + 5 V-Signal (Drehrichtung CCW) oder ein 0 V-Signal (Drehrichtung CW) aus.

4.1.1.10 *Stromabsenkung*

Zur Reduzierung der Temperaturentwicklung von Schrittmotor und Leistungsendstufen verfügen Schrittmotor-Endstufen über eine integrierte Phasenstrom-Reduzierung im Stillstand. Dieses Merkmal kann jedoch zu Problemen bei der Bearbeitung im X-Y-Z-Betrieb zweier oder mehrerer Schrittmotorachsen führen.
Sind z. B. während des Fräsbetriebes einer Achse die Schneidkräfte des Werkzeuges höher als die Halte- bzw. Stillstandskräfte des zweiten nicht bewegten Schrittmotor-Achsantriebes, kann diese Achse aus ihrer Ruheposition bewegt werden und einen undefinierbaren Versatz erfahren. Diese ungewollte Eigenschaft kann umgangen werden, indem während der Bearbeitung alle Achsen den vollen Betriebsstrom zur Verfügung gestellt bekommen.
Aus diesem Grunde verfügt die Interfacekarte über einen Steuerausgang zur definierten Aktivierung der Stromabsenkungslogik innerhalb der Endstufen.

4.1.1.11 *Bremse*

Zur Steuerung einer Haltebremse in Schrittmotor-Systemen unterstützt die Interfacekarte ab Version AZ1350/4 die Ansteuerung eines entsprechenden Steuerrelais. So können Magnetbremsen gezielt ein- und ausgeschaltet werden.

In *isel*- Antriebseinheiten werden Magnetbremsen verwendet, die im Ruhezustand aktiv sind. Diese werden nach dem Power-On-Reset der Interfacekarte über ein Steuerrelais mit + 24 V Betriebsspannung versorgt und so geöffnet (inaktiv).
Je nach Applikation kann die Bremse im Direktmodus des CNC-Betriebssystems programmiert werden.

☞ Die Signalausgänge *Takt, Richtung, Stromabsenkung* und *Bremse* sind über einen 20 mA-Leistungstreiber geführt.

4.1.2 Datenspeicher

Zur Speicherung von systembedingten Variablen und programmierten Funktionsabläufen im CNC-Betrieb verfügen die Interfacekarten über ein 32 kB statisches RAM.
Da dieser Speicher nach Wegfall der Versorgungsspannung die gespeicherten Informationen verliert, ist ggf. in Stand-Alone-Applikationen eine Pufferung der Versorgungsspannung des RAM notwendig. Hierzu verfügt die Interfacekarte optional über eine 100 mAh Akku mit 3,6 V Ausgangsspannung. Ein spezieller Schaltkreis überwacht das Unterschreiten der Versorgungsspannung 4,75 V und sperrt ggf. den Prozessor durch einen Reset-Signal.

5 Optionen und Erweiterungen

5.1 Aufrüstmöglichkeiten

UI 4.0	—>	UI 4.0-E/A	Art.Nr. 328010
UI 4.0	—>	UI 5.0	Art.Nr. 328020
UI 4.0	—>	UI 5.0-E/A	Art.Nr. 328030
UI 4.0-E/A	—>	UI 5.0-E/A	Art.Nr. 328040
UI 5.0	—>	UI 5.0-E/A	Art.Nr. 328050
ab Version UI 4.0	—>	UI 5.C-E/A	Art.Nr. 325551

5.2 Optionen

Programmwahleinheit	Art.Nr. 318110
Akku zur RAM-Pufferung	Art.Nr. 328120
Hand-Terminal UHT1	Art.Nr. 328200
Memory-Card Datenspeicher 32 kByte	Art.Nr. 440114

6 E/A-Erweiterung

Die *isel*-E/A-Erweiterung ist ein Zusatzprodukt zur Interfacekarten-Serie und rundet mit ihren Funktionsblöcken den Bereich 'Schrittmotorantriebe in der Automatisierungstechnik' ab.
Sie erweitert den Funktionsumfang der Prozessorkarte um acht Signalein- und 16 Signalausgänge sowie um einen austauschbaren Datenspeicher (Memory-Card).

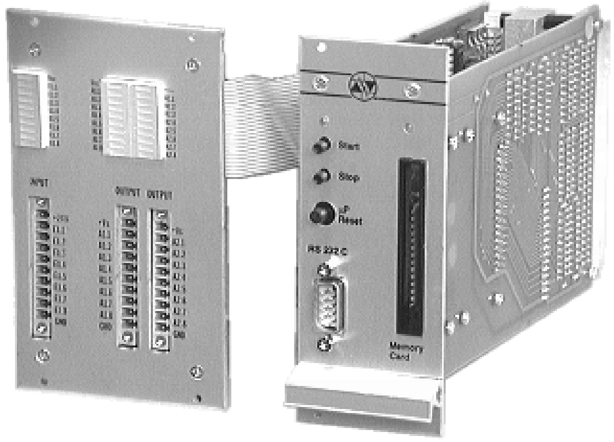


Bild 7: E/A-Erweiterung (montiert auf Interfacekarte und Signalankopplung)

Die E/A-Erweiterung besteht aus einer 100 x 160 mm großen Baugruppe zur Signalverarbeitung und einem Signal-Ankopplungsmodul. Während die Signalverarbeitung direkt mit der Interfacekarte verbunden ist, verfügt die Signalankopplung über eine eigene Frontplatte.

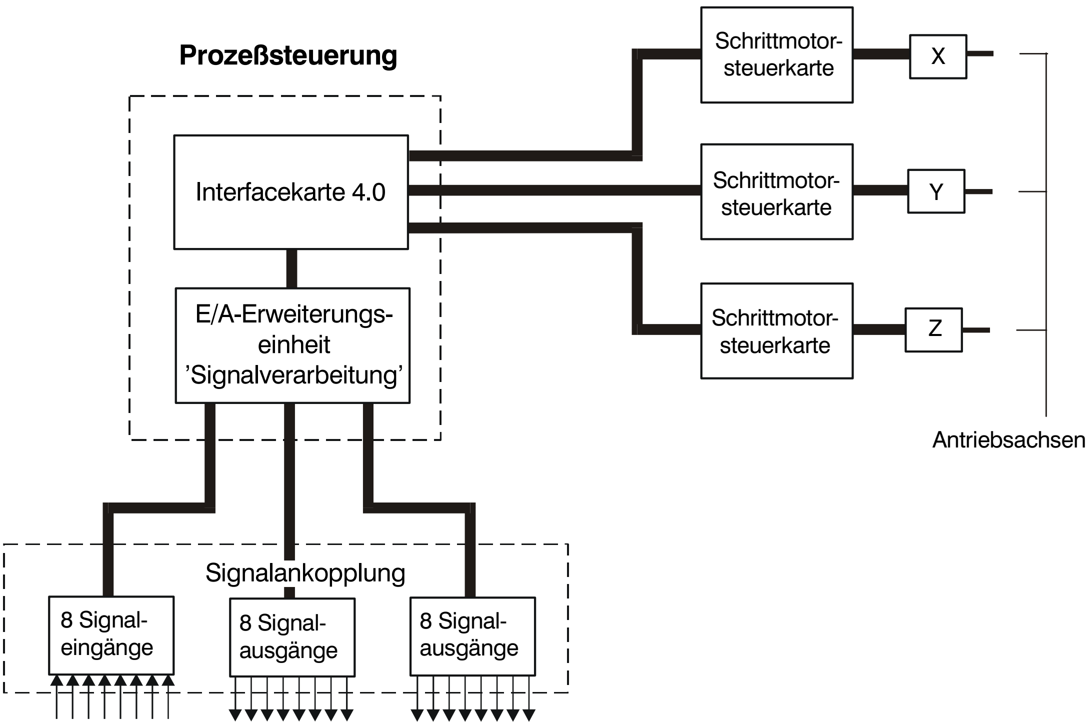


Bild 8: Funktionsblöcke der E/A-Erweiterungseinheiten

6.1 Steckerleiste

Zur Adaption in 19"-Systemgehäusen verfügt die Erweiterungseinhiet über eine 64-polige Stiftleiste nach DIN 41612 Bauform C.

Reihe A			Reihe C		
Signal		Pin	Pin		Signal
GND		1	1	E	GND
NC		2	2		NC
Vcc (+ 5 V)		3	3	E	Vcc (+ 5 V)
NC		4	4		NC
NC		5	5	E	In 1.1
NC		6	6	E	In 1.2
NC		7	7	E	In 1.3
NC		8	8	E	In 1.4
NC		9	9	E	In 1.5
NC		10	10	E	In 1.6
NC		11	11	E	In 1.7
NC		12	12	E	In 1.8
NC		13	13		NC
NC		14	14	A	Out 1.8
NC		15	15	A	Out 1.7
NC		16	16	A	Out 1.6
NC		17	17	A	Out 1.5
NC		18	18	A	Out 1.4
NC		19	19	A	Out 1.3
NC		20	20	A	Out 1.2
NC		21	21	A	Out 1.1
NC		22	22		NC
NC		23	23		NC
NC		24	24	A	Out 2.8
NC		25	25	A	Out 2.7
NC		26	26	A	Out 2.6
NC		27	27	A	Out 2.5
Reset	E	28	28	A	Out 2.4
NC		29	29	A	Out 2.3
NC		30	30	A	Out 2.2
NC		31	31	A	Out 2.1
GND		32	32		GND

NC= nicht belegt
A = Signalausgang
E = Signaleingang

6.2 Signalankopplung

Die Signalankopplung ermöglicht den einfachen Anschluss von externen Sensoren, Relais, elektromagnetischen Ventilen etc. über Schraub-Klemm-Steckverbinder. Die notwendige Versorgungsspannung von + 24 V ist extern zur Verfügung zu stellen und an den Klemmen + 24 V bzw. GND anzulegen.

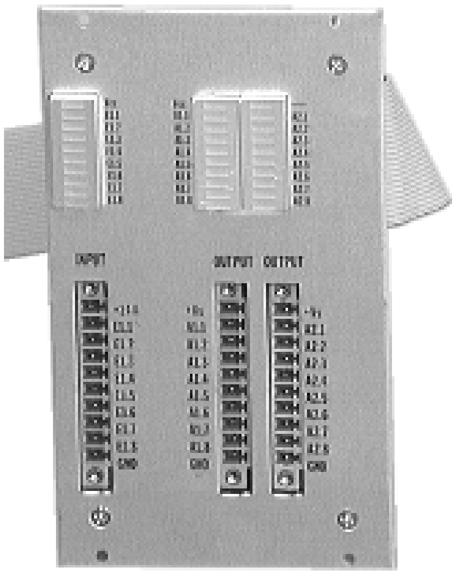


Bild 9: Signalankopplung

6.2.1 Signaleingänge

Die E/A-Erweiterung stellt dem Anwender 8 optoisolierte Signaleingänge zur Verfügung. Entsprechend nachfolgender Zeichnung sind die Eingänge mit einer 12 V-Z-Diode sowie einem Vorwiderstand beschaltet. Hieraus ergibt sich eine Signaleingangsspannung von + 24 V.

Zur optischen Kontrolle der belegten Eingänge stehen LED´s zur Verfügung.

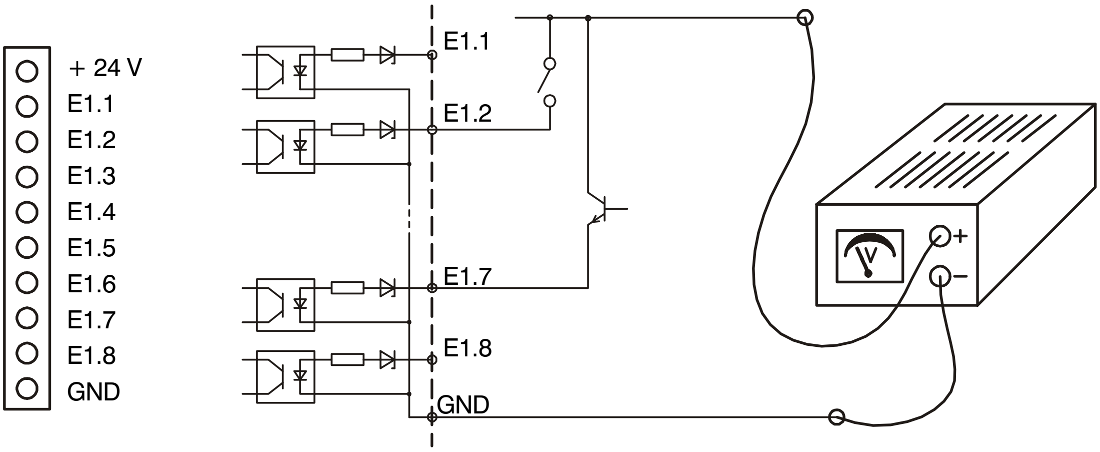


Bild 10: Signaleingänge der E/A-Erweiterung

Die Verarbeitung der Eingänge erfolgt über das Auslesen der Portadresse (65531). Hierzu stehen der Interfacekarte sowohl im DNC- als auch im CNC-Modus entsprechende Befehle zur Verfügung.

DNC-Modus @0b65531
(Auslesen des Eingangsports, byteweise)

CNC-Modus 0 65531 5 1 3

- 3: Vorwärtssprung um 3 Befehlszeilen
  (- 5 = Rückwärtssprung um 5 Zeilen)
- 1: Abfrage ob Signaleingang aktiv
  (1 = Signaleingang aktiv)
  (0 = Signaleingang inaktiv)
- 5: Abfrage Signaleingang 5
- 65531: Adresse des Signaleinganges
- 0: Befehlswort ’Lesen’

6.2.2 Signalausgänge

Die Signalausgänge der E/A-Erweiterung sind als Relais-Schaltausgänge ausgeführt. Die dabei verwendeten Relais erlauben eine maximale Belastung von 50 V bei 300 mA Laststrom. Bedingt durch die 8-Bit-Speicherstruktur der Interfacekarte sind die 16 Ausgänge in zwei 8-bit-Ports unterteilt. Die jeweiligen Port-Adressen sind:

Port A1.1 ... A1.8 Adresse 65529
Port A2.1 ... A2.8 Adresse 65530

Zur optischen Kontrolle verfügt die Signalankopplung über LED-Balkenanzeigen, die bei gesetztem Ausgang leuchten.

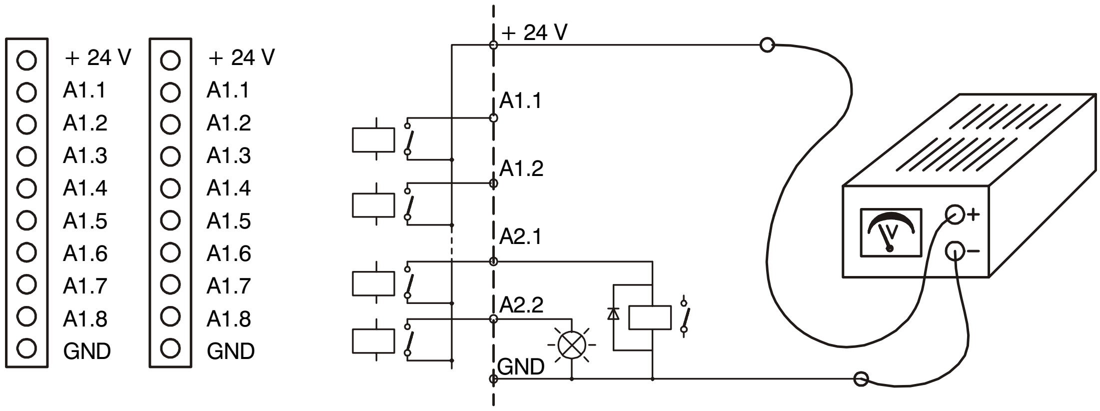


Bild 11: Signalausgänge der E/A-Erweiterung

Die Verarbeitung der Ausgänge wird von der Interfacekarte entsprechend ihrer Programmierung entweder bit- oder byteweise vorgenommen.

DNC-Modus @0b65529,16
(Setzen des Ausgangsports 1 mit dem Binärwert 16)

@0b65530,128
(Setzen des Ausgangsports 2 mit dem Binärwert 128)

CNC-Modus p 65530 5 1

1: Ausgang setzen
(1 = Signalausgang setzen)
(0 = Signalausgang löschen)

5: Ausgang Bit 5

65530: Adresse des Signalausganges

p: Befehlswort 'Schreiben'

6.3 Externer Datenspeicher

Zur externen Speicherung eines Datenfeldes unterstützt die Interfacekarte in Verbindung mit der E/A-Erweiterung den Einsatz eines Scheckkarten-Speichers.
Die Speicherkarte (Memory-Card) mit 32 kB RAM-Speicher und integrierter Batterie wird durch den Befehl @0u mit dem kompletten Inhalt des Interfacekarten-RAM geladen und kann jederzeit durch Betätigen des frontseitigen µP-Reset-Tasters in das RAM zurückgeschrieben werden.

7 Software-Treiber I5DRV

Im Lieferumfang der Interfacekarte ist ab der Version UI5.C die Diskette *isel*-I5DRV enthalten. Dieser Softwaretreiber, der nach dem Laden resident im Hauptspeicher des Steuerrechners bleibt und ab diesem Zeitpunkt für Sie solche Arbeiten wie z. B. die Interpolation und die Kommunikation der Achsenbewegungen, die Verwaltung des Systems, die Kommunikation mit der Hardware etc. übernimmt.
Die Funktionalität des Treibers wird in einer gesonderten Beschreibung *'isel*-Treiber für *isel*-Interfacekarte' behandelt.

Die Beschreibung ist ebenfalls im Lieferumfang enthalten.

☞ Die Programmierung der Interfacekarte mittels der Software PAL-PC ist durch die zusätzliche Funktion nicht eingeschränkt, d. h. bereits erstellte Programme für die Interfacekarte sind voll lauffähig.

isel Power Block 300-C

isel Power Block 450-C

isel Power Block 600-C

Hardware Manual

B.308059/2000.11/E

On this Manual

Various symbols are used in this Manual to quickly provide you with brief information.

Danger	Caution	Note	Example	Additional Information

In spite every care, printing errors and errors can not be excluded.
We welcome any suggestions and remarks on faults.

isel machines and controllers are CE-coforming and adequately labeled. Commissioning of all other machine components is not allowed until all corresponding demands, on which the CE-safety guidelines have to be applied, are fulfilled.

***isel**automation* assumes no guarantee on machines that have been altered or modified.

The electromagnetic compatibility test only applies to the original configuration of the machine supplied ex works.

Manufacturer: Co. ***isel**automation* KG
In Leibolzgraben 16
D-36132 Eiterfeld

Fax: +49-6672-898-888
E-Mail: automation@isel.com
http://www.isel.com

Contents

1 Introduction 4

2 Scope of Supply 4

3 Safety Notes 5

4 Technical Specifications 7

4.1 Motor Voltage 7

4.2 Auxiliary Voltage I 7

4.3 Auxiliary Voltage II 7

4.4 Safety Devices 7

5 System Description 8

5.1 Functional Groups 8

5.2 Connection and Cabling 8

5.2.1 Connector X1 8

5.2.1.1 Signal Outputs (O) 9

5.2.1.2 Signal Inputs (I) 10

5.2.2 Connector X2 10

5.2.3 Connector X3 12

5.3 Status Displays of the Power Block 13

5.4 Coding Field 14

5.5 Fuses 15

5.6 Voltage Output AC 230 V/50 Hz 16

6 Block Diagram of Model PB xx-C Powerblock 17

7 Circuit Documentation 18

7.1 isel Power Block Safety Circuit 18

1 Introduction

Model PB xxx-C *isel* Power Blocks are rack-mounting units designed especially for the power supply of *isel* power units (CV 4, C 142-4). The steel-sheet-enclosed devices (dimensions W = 150 x H = 140 x D = 220 mm) incorporate a 650 VA toroidal-core current transformer with starting current limitation and mains filter, as well as a p.c. board for providing auxiliary voltages and safety-relevant function elements.
The Power Blocks are offered in three different variants that differ only by the height of the load voltage (supply voltage of power output stages).

PB 600-C	Voltage output 68 V/7 A
PB 450-C	Voltage output 43 V/8 A
PB 300-C	Voltage output 30 V/8 A

Fig. 1: Model PB xxx-C isel Power Block

2 Scope of Supply

The scope of supply of Model PB xxx-C Power Block comprises:

- Power Block with mains supply cable (l = 0.5 m)

3 Safety Notes

- When installing or using the Power Block, please observe the standards laid down in the Certificate of Conformity.
- The instructions and limit values observed by the manufacturer will not provide protection in case of improper use of the device.

In this context, you should observe the following:

... Connect and install the device only when it is turned off and the mains line is removed.

... All work on the device should only be carried out by expert personnel. When doing so, adhere, in particular, to the relevant regulations and instructions of the electrical industry, as well as to the relevant rules for the prevention of accidents.

Relevant standards applicable to the stepper motor controller:

EN 60204 (VDE 0113) Part 1 (1992 Edition)
- Electrical Equipment of Industrial Machines

EN 50178 (VDE 0160)
- Completion of Electrical Power Installations with Electronic Equipment

VDE 0551
- Regulations for Safety Isolating Transformers

EN 292 Parts 1 and 2
- Safety of Machinery

EN 55011 (VDE 0875)
- Radio and Television Interference Suppression, Limit Value B

IEC 1000-4 (Parts 2-5)
- Inspection, Test and Measuring Methods for Noise Immunity

The Power Blocks require the following supply voltages:

PB 600-C	AC 230 V/50 Hz, max. 8 A
PB 450-C	AC 230 V/50 Hz, max. 7.5 A
PB 300-C	AC 230 V/50 Hz, max. 7.0 A

The network transformer has a temperature switch on its primary side, which has a response temperature of 120 °C. When connecting the Power Block, install additional primary fuses. When connecting the device directly to the domestic electrical installation, primary protection of the Power Block is provided by the fuse element (16 A) installed therein.
When integrating the Power Block into a control system (e.g. a control cabinet), an additional primary fuse must be installed. To do so, use exclusively fuses to IEC-127. The mains supply cable is carried into the Power Block via a PG9-type heavy-gauge conduit thread (capacity of terminals: 4 ... 8 mm). The connecting cable must be a double-isolated line.

When installing the Power Blocks, the following considerations should also be observed:

- The Power Block is a rack-mounted unit of class of protection 1.
- The degree of protection of the Power Block is IP 20.
- The installation of the Power Block may only be carried out lying horizontally.
- Primary and secondary lines must be designed as cables (no single lines).
- Primary and secondary lines must be separated by 3 layers of insulating material.
- The Power Block is designed for operation at an ambient temperature of max. 40 °C.

4 Technical Specifications

4.1 Motor Voltage

For supplying the power output stages, Model PB xxx-C Power Blocks provide a non-stabilised DC voltage (DC link voltage). The voltage output is enabled by a safety device with switching relay connected in series.

The safety device constitutes a series connection of control stations which turns on the secondary voltage of the toroidal-core current transformer using a safety relay with positively driven contacts and a series-connected all-or nothing relay. The output voltage of the DC link is connected to WAGO terminals via four separated fuse-elements. The connecting lines connected there are brought off the power supply module via heavy-gauge conduit threads.

To protect the DC link voltage from overvoltage (e.g. by energy recovery in brake mode of the motors), the Power Block is provided with an appropriate protective circuitry (brake chopper). In case of voltages > 80 V, it automatically enables a power resistor converting energy into heat.
When the safety circuit is disabled, the stored energy of the DC link capacitor is discharged via a load resistor.

4.2 Auxiliary Voltage I

This + 24 V auxiliary voltage is provided from the output of a fixed-voltage controller. The input voltage is provided by a double-insulated secondary winding of the transformer.
The + 24 V voltage serves for power supply of the signal inputs/outputs, the external limit switch and reference switches, as well as of the control relay of the safety circuit.
The maximum current that can be used by an external load is 0.7 A.

4.3 Auxiliary Voltage II

This + 24 V auxiliary voltage is intended for power supply of the safety circuit. The input voltage is provided from a double-insulated secondary pick-off of the toroidal-core current transformer. The + 24 V fixed-voltage controller is used to limit the output current to approx. 1.0 A.

4.4 Safety Devices

The implementation of the safety circuit is based on a series connection of control stations, e.g. EMERGENECY STOP switch, safety loops and ON button.
The safety-relevant parts act on a relay with positively driven contacts that, in turn, turns on load relays. The load relays are monitored by an opto-coupler acc. to EN 60204 and are designed redundantly.

5 System Description

5.1 Functional Groups

① Auxiliary voltage 1

② Auxiliary voltage 2

③ Four load relays

④ Secondary. conn. 2

⑤ Secondary. conn. 3

⑥ Connection terminal X3

⑦ Safety relay

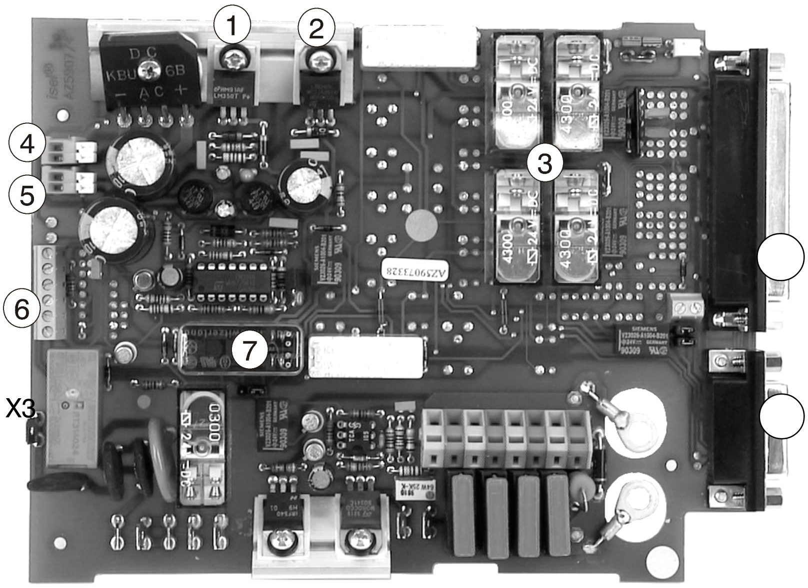


Fig. 2: Functional groups of the Power Block

5.2 Connection and Cabling

5.2.1 Connector X1

For connecting the power units, the Power Block has a 37-pin Sub D female connector.

Signal		Pin	Pin		Signal
GND	**A**	1	20		Not connected
+ 24 V	**A**	2	21		Not connected
Not connected		3	22	**E**	Limit switch
Power output stage disable 1	**A**	4	23	**A**	Limit switch enable
GND	**A**	5	24	**E**	Drive enable
+ 24 V	**A**	6	25		Not connected
Not connected		7	26		Not connected
Power output stage disable 2	**A**	8	27	**A**	Safety circuit o.k.
GND	**A**	9	28		Not connected
+ 24 V	**A**	10	29		Not connected
Not connected		11	30		Not connected
Power output stage disable 3	**A**	12	31	**A**	Load relay monitoring
GND	**A**	13	32		Not connected
+ 24 V	**A**	14	33		Not connected
Not connected		15	34		Not connected
Power output stage disable 4	**A**	16	35		Not connected
Not connected		17	36	**A**	Opto-coupler X1
GND	**A**	18	37	**A**	GND
GND	**A**	19			

5.2.1.1 Signal Outputs (O)

+ 24 V, GND

This is the voltage output of auxiliary voltage I.
The voltage is intended to supply the reference switches of the numerical axes and the opto-couplers in the power electronic unit.

Power output stage disable

This output is intended for disabling (making dead) the stepper motor power output stages. The + 24 V output signal of the Power Block is enabled with the load relay disabled (supply voltage of power output stages switched off) and provided to all connected output stage boards in parallel.

Limit switch enable

The signal output provides a +24 V signal, with the limit switch monitoring circuit by-passed.

By-passing of this safety-relevant functional group is necessary if one or several limit switches are active. This may be caused, e.g. by a mistake in the drive unit (controller) or mechanics or due to faulty operation (see also „Signal Output Enable“).

The signal is only used in conjunction with servomotor power units.

Safety relay o.k.

A + 24 V voltage on connector X1.27 signals that the power supply of the power output stages (DC link voltage) is turned on.
This output signal is the control voltage of the load relays and connected to the plug-in contact; pin X1.9 (GND) is used as the earth reference.

Monitoring load relay

This is a + 24 V output (open emitter) of an opto-coupler that monitors the signals provided from the switching contacts of the redundantly designed load relays. With the DC link voltage switched off (safety relay is disabled) and defective load relays (e.g. contact is welded), the output carries + 24 V potential.

Opto-coupler X1

The output is the status display of the safety circuit. The opto-coupler output is enabled (+ 24 V connected) if the series connection of the safety-relevant controls is operative so that actuating the ON button results in switching the safety relay.

5.2.1.2 Signal Inputs (I)

Limit switches

Limit switches installed on the numerical axes are intended to limit the maximum traversing distances. They are effective directly in the safety chain of the Power Block via relay and, when actuated, interrupt the power boards connected.

For activating the relay, a +24 V signal should be connected to the input of the power block. If the control voltage is not provided, the relay will drop out, interrupting the safety device.

The limit switch signal input is only evaluated in servomotor power units. All limit switches on the interface module of the controller are monitored and carried as a group signal to the Power Block.

Drive enable

For monitoring the readiness for operation of the connected power units or of a control computer, the Power Block expects an enable signal.
The + 24 V signal is effective in the safety circuit of the Power Block via a relay.

5.2.2 Connector X2

The 15-pin Sub D-female connector X2 is prepared for connecting external, safety-relevant controls. EMERGENCY STOP switch, ON button, safety contacts, etc. can be connected here acc. to the assignment below.

Signal	Pin	Pin	Signal
Key switch (n. o. contact)**	1	9	Key switch (n. o. contact)**
ON button (n. o. contact)	2	10	ON button (n. o. contact)
Safety switch (n. o. contact)	3	11	Safety contact (n. c. contact)
Safety contact (n. c. contact)	4	12	Safety contact (n. c. contact)
EMERGENCY STOP switch (n. c. contact)	5	13	EMERGENCY STOP switch (n. c. contact)
Safety relay (GND)	6	14	Safety relay (+ 24 V = enabled)
Potential-free switching contact	7	15	Potential-free switching contact
Load relay monitoring	8		

** is only evaluated in conjunction with servomotor power units

The pin assignment is as follows:

Key switch

A closed contact between X2.1 and X2.9 jumpers the limit switch monitoring. As a result, an actuated limit switch on the numerical axes will not turn off the operating voltage (see also Section 4.2.1.2: Limit Switch Enable).

When using a key switch, make sure that the contact is turned on not longer than absolutely necessary.

The protective devices of the drive axes are disabled! It is imperative to observe the maximum traversing distances of your drive axis. In case of a collision within the mechanics, impairments of the functional performance cannot be ruled out.

ON button

A normally open contact (n. o. contact) between X2.2 and X2.10 will turn on the DC link voltage (power supply of power output stages) if all function elements of the safety chain are active.

Safety switch

Actuating a normally closed (n. c.) switch connected between the contacts X2.3 and X2.11 will turn off the operating voltage of the output stages.
The switch should be chosen and used acc. to EN 418. If the switching contact is not needed, the contacts should be jumpered.

Safety contact

Actuating a normally closed (n.c.) switching contact connected between the contacts X2.4 and X2.12 will turn off the operating voltage.
The switch should be chosen and used acc. to EN 418. If the external switching contact is not needed, the contacts should be jumpered.

EMERGENCY STOP switch

Actuating an n.c. switch of an EMERGENCY STOP switch connected between the contacts X2.5 and X2.13 will turn off the operating voltage. The switch should be chosen and used acc. to EN 418. If the EMERGENCY STOP switch is not connected, the contacts should be jumpered.

Safety relay active

The output is the control voltage of the load relay installed in the Power Block. Thus, a +24 V voltage is present at output X2.14 when the load relay is enabled; earth reference is contact X2.6.

Potential-free switching contact

The outputs X2.7 and X2.15 are connected to a potential-free relay contact within the Power Block. The contact is closed when the DC link voltage (operating coltage of the power output stages) is turned on. It can be used to integrate the Power Block into higher-level safety systems.

Load relay monitoring

With the safety circuit disabled and defective switching contacts of the all-or-nothing relay, this output will provide a + 24 V signal (pulsating DC voltage). Contact X2.6 is used as the earth reference.

5.2.3 Connector X3

The following controls can be connected to the 8-pin board terminal:

1 - 2 EMERGENCY STOP switch
3 - 4 ON button
5 - 6 ON button lamp (lights when the safety circuit is enabled)
7 - 8 Key switch

☞ The function of the switching elements is identical to that of the 15-pin Sub D male connector X2. When connecting the ON pushbutton, make absolutely sure that only one ON switching function may exist acc. to the relevant safety standards and thus a second ON pushbutton may not be connected externally to connector X2 at the same time.

5.3 Status Displays of the Power Block

To display the operating states, the Power Block has four LEDs (V1 to V4)

① LED V1

② LED V2

③ LED V3

④ LED V4

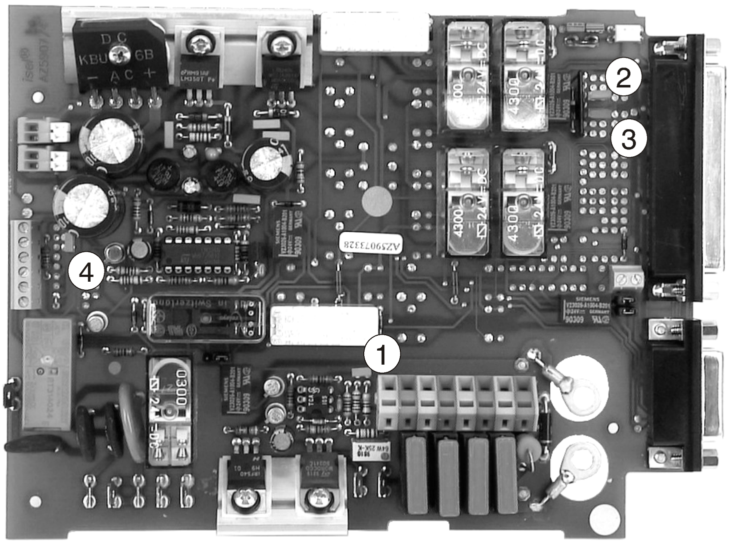


Fig. 3: LEDs of Power Block PB xxx-C

The LEDs are:

V1 LED V1 displays that the brake chopper is active, thus connecting a load resistor in parallel to the DC link capacitor. This operating state can be achieved as a result of two events:

- Overvoltage on the capacitor, e.g. by energy recovery from the connected DC servomotor.
- Connecting the load resistor for fast reduction of the stored energy of the DC link capacitor after disabling the safety circuit.

V2 LED V2 signals that the limit switch input is active (+ 24 V) and monitoring by the safety circuit is provided.

V3 LED V3 lights when a +24 V signal is present at the drive enable signal input and the power output stages thus signal their readiness for operation. The input is effective directly in the safety circuit of the Power Block.

V4 LED V4 lights when the safety circuit is ready for operation, i.e. the following safety-relevant controls are active.

- EMERGENCY STOP switch (external) N. C. CONTACT
- Safety switch (e.g. cover contact) N.C. CONTACT
- Emergency stop switch (internal)
- Safety contact (e.g. kick-strip, lighting trunking) N.C. CONTACT

5.4 Coding Field

The control board of the Power Block has three coding jumpers that can be used to adapt the Power Block to different operating conditions.

Jumper 2 **Jumpers 1 and 3**

Fig. 4: Coding fields and fuses of the Model PB 600-C Power Block

Coding fields on J1

Coding plug J1 is intended to prepare the connection of an external ON button to connector X2. Since acc. to the Machine Protection Regulations the DC link voltage of the Power Block may only be connected using on ON button, after connecting jumper J1 make absolutely sure that the ON button can no longer be operated (removing the pushbutton lines from connector X3, covering the actuating knob, etc.).

Coding jumper J2

This coding jumper can be used to determine the switching time of the output relay (see Section 4.6). You can choose between two operating states:

- **J2.1** The output relay will switch at the same time with the load relay of the Power Block. In this case, the output voltage (AC 230 V/50 Hz) is enabled by the safety relay.
- **J2.2** If coding jumper '2' is connected, the output relay will switch immediately after turning on the transformer.

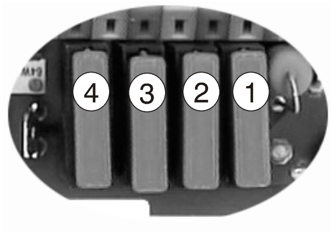
Coding jumper J3

This coding jumper is intended for extensions (if intended) and closed on Power Block PB xxx-C.

5.5 Fuses

Fuses F1 - F4

The DC link voltage is picked off from four separated WAGO terminal blocks. A maximum of four power output stages can be connected.
To protect the voltage output, a fuse (F1 - F4) is connected in series to each terminal block.
These are FKS-type fuses with a nominal value of 5 A (sluggish).



Fuses F1 - F4

Fuses F5 and F6

The fuses F5 and F6 are connected in series in the output line of the all-or-nothing relay (see Section 4.6), protecting the relay from overload. These are two fusible links to IEC-127 with a nominal value of 4 A (sluggish).

Temperature switch T1

Temperature switch T1 is connected in series in the primary winding of the tcoroidal-core current transformer. The temperature switch responds if the temperature exceeds 120 °C. After the transformer has cooled down to approx. 60 °C, the temperature switch is turned on automatically. The self-holding feature of the safety relay guarantees that the motor voltage is not enabled.

☞ Since the primary circuit of the mains transformer is protected from overload merely with a temperature switch, you must install an additional primary fuse when installing the Power Block. When connecting the Power Block directly to the domestic electrical installation, a primary protection of the Power Block by fuses is provided by the fuse element (16 A) installed in the Power Block.

☞ When integrating the Power Block into a control system (e.g. control cabinet), an additional primary fuse must be installed.

☞ Use exclusively fuses to IEC-127. The fuses with a nominal value of 8 A should have a sluggish switching response.

Temperature switch T2

Temperature switch T2 is connected in series in secondary winding 2 (auxiliary voltage II) of the mains transformer. The response temperature of the switch is 120 °C.

Temperature switch T3

Temperature switch T3 is connected in series of secondary winding 3 (auxiliary voltage I) of the mains transformer. The response temperature of the switch is 120 °C.

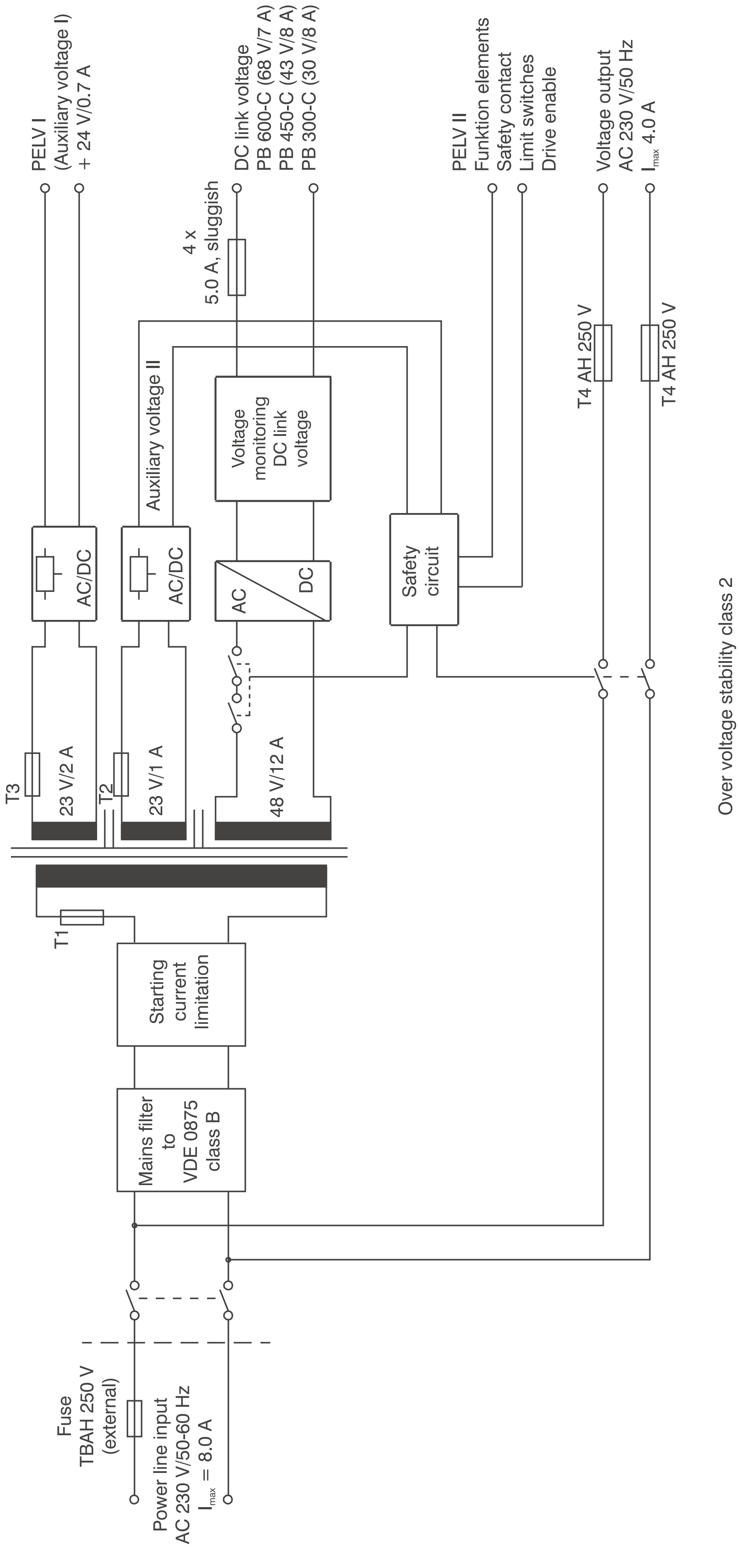
5.6 Voltage Output AC 230 V/50 Hz

For controlling an additional external device (input voltage AC 230 V/50 Hz , max. 4 A), the Power Block provides an appropriate output.
This output is connected by a load relay electronically coupled with the safety relay of the safety circuit. As a result, the output voltage is only available if all safety-relevant parts are enabled.

The voltage output is protected by two fuses T 4.0 A H 250 V (5 x 20 mm, IEC-127). A 3-core PVC-insulated line brought out from the housing via a PG-11 heavy-gauge conduit thread is used for connection.

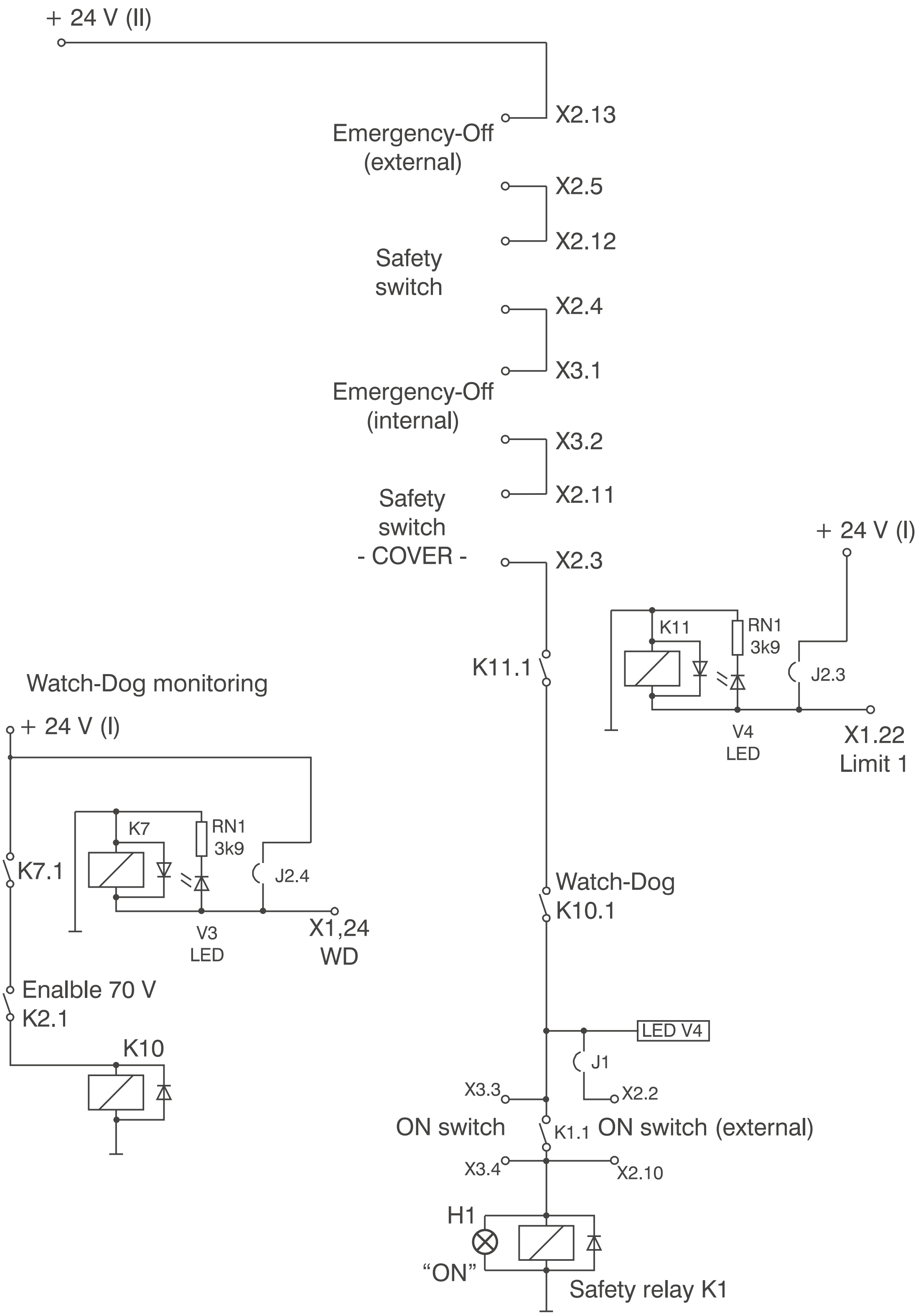
If the output cable is connected later, a double-insulated line (no single lines) with a cross-sectional area of 1.0 mm should be used. Due to the PG-9 heavy-gauge conduit thread, the cable diameter should be within a range of 4 ... 8 mm.

6 Block Diagram of Model PB xx-C Powerblock



7 Circuit Documentation

7.1 *isel* Power Block Safety Circuit



iselautomation KG

isel-CNC Operating System 5.x

Software manual

970325 BE003
10/2000

On this Manual

Various symbols are used in this Manual to quickly provide you with brief information.

Danger	Caution	Note	Example	Additional Information

Despite all care, printing errors and mistakes cannot be ruled out completely.
Suggestions for improvement and notes on errors are always welcomed.

Manufacturer: ***isel***automation KG
Im Leibolzgraben 16
D-36132 Eiterfeld

Fax: (06672) 898-888
e-mail: automation@isel.com
http://www.isel.com

Contents

1	**Introduction**	**.5**
2	**DNC Command Structure**	**.6**
2.1	Basic Command Set for Processor Card 4.0 (and higher)	.7
2.1.1	Command: Set number of axes	.7
2.1.2	Command: Reference point approach	.8
2.1.3	Command: Set reference speed	.10
2.1.4	Command: Relative movement	.11
2.1.5	Command: MoveTo (position)	.13
2.1.6	Command: Position interrogation	.15
2.1.7	Command: Zero offset	.16
2.1.8	Command: Select plane	.17
2.1.9	Command: Peek (read memory address)	.19
2.1.10	Command: Poke (write memory address)	.20
2.1.11	Command: Clear battery-backed RAM	.21
2.1.12	Command: Set CR/LF	.22
2.1.13	Command: Set device number	.23
2.1.14	Command: TRACE (single-step mode)	.24
2.1.15	Command: Self-test	.25
2.2	Supplementary Command Set of Interface Card 5.0	.26
2.2.1	Command: 3D linear interpolation	.26
2.2.2	Command: Circular interpolation	.28
2.3	Supplementary Command Set of Interface Cards with I/O Expansion	.34
2.3.1	Command: Save externally	.34
2.3.2	Command: Set output port	.35
2.3.3	Command: Read input port	.35
2.4	Supplementary Command Set of EP1090	.36
2.4.1	Command: Output module	.36
2.5	Supplementary Command Set for Interface Card, Version AZ1350/5 and Higher	.36
2.5.1	Command: Magnetic brake	.36
2.6	Check and Control Codes	.37
2.6.1	Command: Self-test	.37
2.6.2	Command: STOP	.38
2.6.3	Command: µP Reset	.39
2.6.4	Command: Break	.39

...

3 CNC Command Structure.40

3.1 Basic Command Set of Processor Card 4.0 and Higher 41

3.1.1 Command: INPUT 41
3.1.2 Command: Reference Point Approach 42
3.1.3 Command: Relative Movement 43
3.1.4 Command: MoveTo (position) 44
3.1.5 Command: Zero offset 45
3.1.6 Command: Select plane 46
3.1.7 Command: Transmit synchronisation character 47
3.1.8 Command: Wait for synchronisation character 49
3.1.9 Command: Loop / Branch 50
3.1.10 Command: Pulse Control 52
3.1.11 Command: Time Delay 53
3.1.12 Command: Move to pulse 54
3.1.13 Command: Start connected interface card 55

3.2 Supplementary Command Set of Interface Card 5.0 56
3.2.1 Command: 3D Linear Interpolation 56
3.2.2 Command: Circular interpolation 57

3.3 Supplementary Command Set of Interface Cards with I/O Expansion 59
3.3.1 Command: Set output port 59
3.3.2 Command: Read input port 61

3.4 Supplementary Command in Conjunction with a Program Selection Unit 62
3.4.1 Command: Keyboard polling 62

4 Error Messages 64

4.1 Error Messages of the Processor Cards 64

4.2 PAL-PC Error Messages 68

1 Introduction

The description of the CNC operating system 5.x is a comprehensive documentation of all commands of ***isel*** processor cards. The commands described herein apply to the following ***isel*** control systems:

- ***isel*** Interface Card (up to software version 5.x)
- ***isel*** CNC Controllers C 116, C 142/1, C 116-4, C 142-4
- ***isel*** CNC Control Systems C 10C, C 10C-I/O
- ***isel*** Integrated Technologies IT 108, IT 116
- ***isel*** Machining Centre EP 1090
- ***isel*** Machining Centre EP 1090/4

The CNC operating system supports the positioning of a maximum of three stepper motor drive axes. In addition to the positioning parameters, the operating system is able to process various control and check functions.

Due to the fact that all control systems are summarised in one operating system (called here processor card), certain restrictions regarding the programming of the individual devices may possibly be taken into account. These restrictions are mentioned in the relevant hardware descriptions.

The program examples used in the Description refer to the maximum configuration. In some cases, it may be therefore necessary to adapt the positioning commands accordingly to the particular application.

The term ‘PAL PC’ is used both in conjunction with the programming language PAL-PC and with the PAL-EP software interfacing module.

For direct programming of the processor cards, a defined transmission format is provided. This Manual contains an example programmed in BASIC.

2 DNC Command Structure

In DNC mode, data records and commands transferred from a control computer are evaluated and executed directly. To this aim, a so-called initialisation is required prior to the data communication. This initialisation consists of the data opening character @, the device number (default = 0) and the number of axes to be traversed. Thereafter, the program steps are transferred to the processor card separately and executed directly. For checking the data transfer and providing appropriate messages in case of errors, ASCII characters are sent back to the control computer via the interface. This so-called hardware handshake procedure can be realised at two different times:

1. The processor card will send off the acknowledgement/error flag directly after receiving the data record to be executed.
2. The processor card will execute the transmitted command set and will then feed back the acknowledgement character/error flag.

The desired mode is distinguished by the use of capital/small letters for the command character. If capital letters are used, a check-back signal is provided after the respective command has been executed, and small letters will result in a direct check-back signal.

The command set of Interface Card 4.0 is described in the following. For amendments resulting from hardware upgrades (e.g., Interface Card 5.0), please refer to the end of the Chapter.

The terminal mode mentioned in the example programs is a function of the ***isel*** PAL-PC software. It is enabled in PAL-PC using function key F2 and provides a direct link between screen and interface card.

For further information, please refer to the PAL-PC Manual, Section X1, “Communication Window“.

2.1 Basic Command Set for Processor Card 4.0 (and higher)

2.1.1 Command: Set number of axes

Application The processor card is re-initialised by transmitting the number of axes. The data memory will be cleared and, to optimise the memory, re-allocated according to the number of axes.

Structure <GN><axes>

<GN> = device number, default = 0

<axes> = axis specification, see below

Notation @07

Explanation The card is addressed using @0; the axis configuration is specified by the numerical value after the address.

Axis specification	Value
x	1
xy	3
xz	5
xyz	7

Restrictions The combinations @00, @02, @04, @06, as well as @08 and @09 are not allowed.

Programming example

<u>PAL-PC</u>	<u>GW-BASIC</u>
#axis xyz;	100 open“com1:9600,N,8,1,DS,CD“as #1

```
100 open“com1:9600,N,8,1,DS,CD“as #1
110print#1,“@07":gosub 1000
120 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$=“0" then return
1020 print “card signals error“: ;a$
1030 stop
```

The command *Set number of axes* will clear all data stored in the RAM, even if the data have been stored in the RAM of the processor card thanks to the integrated option *Memory Back-up* after a failure of the supply voltage.

2.1.2 Command: Reference point approach

Application The processor card will traverse all specified axes to their zero points (reference points). With ***isel*** systems, the reference points are located in the direction towards the motor; positive movements will be carried out in the direction from the motor away.

Structure @<GN>R<axes>

<GN>	=	device number, default = 0
<axes>	=	integer value between 1 and 7

Notation @0R7 or @0r7

Explanation The card is addressed using @0. “R“ specifies that approach to the reference points is to be carried out. The numerical value defines the axes to be referenced:

x	=	1	xy	=	3
y	=	2	yz	=	6
z	=	4	xyz	=	7
xz	=	5			

The order of execution is defined as follows:

—> Z axis —> Y axis —> X axis

This will also be true if an axis other than the tool axis has been defined using the Plane command. In this case, any collisions with the workpiece can be prevented if the individual axes are approached to their reference points separately.

After the reference point approach has been carried out, the processor card will send its acknowledgement flag and will wait for the commands to come. If an immediate check-back signal is required, use “r“ instead of “R“. The processor card, however, can execute commands only after the mechanical system has carried out the reference approach.

Restrictions You can use this command after an initialisation of the processor card has been carried out using the command *Set number of axes;* the command is limited to the speficied axis configuration. In case of incorrect axis specification, error check-back signal 3 will be provided. If the card is in 3D mode, this command will switch the card back to 2.5D mode.

Programming example

PAL-PC	GW-BASIC
#axis xyz;	100 open“com1:9600,N,8,1,DS,CD“ast#1
reference xyz;	110 print#1,“@07":gosub 1000
	120 print#1,“@0R7":gosub 1000
	130 stop
	1000 if loc(1)<1 then goto 1000
	1010 a$=input$(1,1)
	1015 if a$=“0" then return
	1020 print ”card signals error:”;a$
	1030 stop

If the reference switch is not connected, the processor card will provide pulses continuously. By pressing the STOP key twice, however, you can abort reference point approach of the axis concerned.

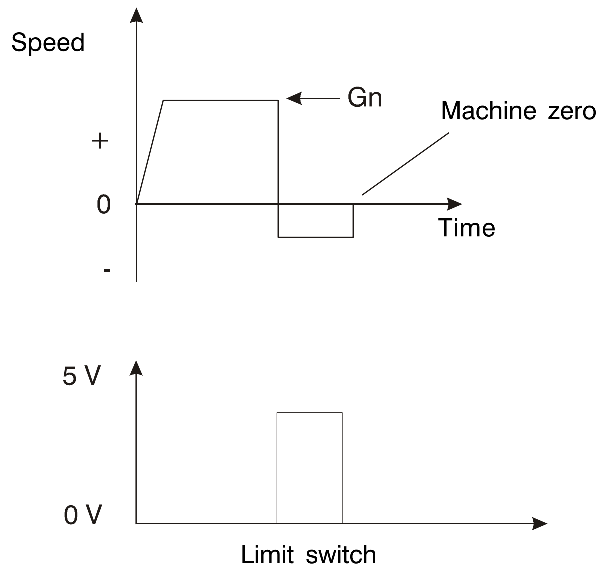


Fig. 1: Course of reference point approach

2.1.3 Command: Set reference speed

Application This command is used to define the speed at which referencing is carried out for each axis seperately. This is only the speed at which the axis approaches to the motor in the negative direction; the speed from the switch cannot be controlled (see “Reference point approach“).

Structure

```
@<GN>d<Gx>                 (x)
@<GN>d<Gx>, <Gy>           (X-y)
@<GN>d<Gx>, <Gz>           (X-z)
@<GN>d<Gx>, <Gy>, <Gz>     (X-Y-z)
```

<GN>	=	device number, default = 0
<Gx>	=	referencing speed x
<Gy>	=	referencing speed y
<Gz>	=	referencing speed z
<Gx>, <Gy>, <Gz>	=	integer number between 30 and 10,000 Hz

Notation

```
@0d2500              (1 axis)
@0d2400,3000         (2 axes)
@0d1000,3000,9000    (3 axes)
```

Explanation If no information on the referencing speed is transferred to the processor, the reference points will be approached at a default speed of 2,000 steps/s. Any modifications to the values remain stored when the device is switched off provided the *Memory Back-up* option is installed.

Restrictions -

Programming example

PAL-PC

```
#axis xy;
#ref_speed  3000,5000;
```

GW-BASIC

```
100open“com1:9600,N,8,1,DS,CD“as #1
110 print#1,“@03":gosub 1000
120 print#1,“@0d3000,5000":gosub 1000
130 print#1,“@0R3":gosub 1000
140 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$=“0" then return
1020 print ”card signals error:”;a$
1030 stop
```

A too high referencing speed in conjunction with a high leadscrew pitch may cause damage to the reference switches due to the existing mass inertia.
The processor card requires a switching hysteresis of the connected zero-position switch (to be observed when connecting electronic zero sensors).

2.1.4 Command: Relative movement

Application The processor card will provide a pulse sequence according to the transferred number of steps and according to the step speed for each power output stage. The traversing movement will either be carried immediately or will be stored.

Structure @<GN>A<Sx>,<Gx>,<Sy>,<Gy>,<Sz1>,<Gz1>,<Sz2>,<Gz2>

<GN>	=	device number, default = 0
<Sx>	=	number of steps x, value between 0 and +/- 8,388,607
<Gx>	=	speed x, value between 30 and 10,000
.		
.		
<Gz2>	=	speed of Z axis (2nd movement)

Notation

@0A 5000,900	(only X axis)
@0A 50,900,20,9000	(X and Y axes)
@0A 50,900,20,900,-20,900	(X and Z axes)
@0A 30,800,10,900,4,90,-4,30	(X, Y and Z axes)

Explanation The processor card is addressed using @0; “A“ specifies that a movement is to be carried out. The processor card will now expect a pair of numbers consisting of the number of steps and the speed for each individual axis.

The movement is carried out in incremental dimensions, i.e. with reference to the last position. The specified number must correspond to the number of axes, i.e. one parameter pair for X mode, two parameter pairs for XY mode, three parameter pairs for XZ mode and four parameter pairs for XYZ mode. The individual numbers must be separated by commas.

For the Z axis, two pairs of numbers are expected, since the situation “Travelling, lowering the tool and then lifting“ very often occurs.

In 2.5D interpolation mode, first the X and the Y axes will traverse (with linear interpolation), and then the Z axis will traverse first by the values specified in z1 and then by the values specified in z2. This interpolation assignment can be modified in 2D mode using the Plane command.

If only one axis is to be moved, nevertheless the values for all initialised axes have to be transferred. When doing so, a value between 30 and 10,000 must be specified for the numbers of steps of the axes not moved.

After the command has been executed, the processor card will provide the handshake character (0) as the check-back signal. Making use of the distinction between the different command codes “a“ and “A“, you can choose between an acknowledgement message provided directly after the transmission and a check-back signal provided after the command has been executed. In any case, however, the processor card can only execute commands after a command has been completed.

Restrictions You can use this command only after the number of axes has been set. The processor card will not check whether the movement leaves the admissible range of the connected mechanical system.

Programming example

PAL-PC

```
#axis xy;
move50(500),300(900);
```

GW-BASIC

```
 100 open"com1:9600,N,8,1,DS,CD"as#1
 110 print#1,"@03":gosub 1000
 120 print#1,"@0A50,500,300,900":gosub 1000
 130 print#1,"@0A20,200,-30,900":gosub 1000
 140 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$="0" then return
1020 print "card signals error : ";a$
1030 stop
```

In 2.5D interpolation mode, the speed specification of the axis with the longest travel will be accepted as the traversing rate, and the speed of the other axis will be matched according to the travel ratio.
In contrast to this, in 3D interpolation mode, the speed specification of the X axis will be used as the set value for the traversing rate.

2.1.5 Command: MoveTo (position)

Application The processor card will traverse to the specified position at the specified rates. The traversing movement will be carried out immediately.

Structure @<GN>M<Sx>,<Gx>,<Sy>,<Gy>,<Sz1>,<Gz1>,<Sz2>,<Gz2>

<GN> = device number, default = 0
<Sx> = positional data for X axis
<Gx> = speed of X axis
.
.
<Sz2> = with absolute positioning = 0
<Gz2> = speed of Z axis (2nd movement)

Notation

@0M 5000,900	(X axis)
@0M 50,900,20,9000	(X and Y axes)
@0M 50,900,20,900,0,21	(X and Z axes)
@0M 30,800,10,900,4,90,0,21	(X, Y and Z axes)

Explanation The processor card is addressed using @0. “M“ specifies that an absolute position will follow. For sake of compatibility with the relative positioning command, two pairs of numbers are also here expected for the Z axis. The second position specification, however, must be zero and will be ignored. After the command has been executed, the processor card will send the handshake character as the check-back signal. If you wish to be provided the check-back signal immediately, use “m“ instead of “M“. In any case, however, the processor card can only receive new commands after the execution of this command has been completed.

Restrictions You can only use this command after the number of axes has been set. It cannot be transmitted during the execution of stored commands. The processor card will not check whether the movement leaves the admissible range of the connected mechanical system. To save the command, first set the processor card to Input mode (see „Input“) and use command code “m“.

Programming example

<u>PAL-PC</u>

```
#axis xy;
reference xy;
moveto 50(500),300(900);
moveto 20(200),30(900);
moveto 0(21),00(2000);
stop.
```

<u>GW-BASIC</u>

```
100 open“com1.9600,N,8,1,DS,CD“as #1
110 print#1,“@03":gosub 1000
120 print#1,“@0M50,500,300,900"
125 gosub 1000
130 print#1,“@0M20,200,30,900"
135 gosub 1000
140 print#1,“@0M 0,21,700,2000"
145 gosub 1000
150 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$=“0" then return
1020 print ”card signals error: ”;a$
1030 stop
```

2.1.6 Command: Position interrogation

Application The processor card will feed back the current set position of all axes to the higher-level computer.

Structure @<GN>P
<GN> = device number, default = 0

Notation @0P

Explanation The processor card is addressed using @0. “P“ specifies that a position interrogation is to be carried out. The processor card will confirm this with the handshake character and will then output the position values of all axes in the hexadecimal format (in total, 19 bytes = 18 hexadecimal digits + 1 x handshake)
The structure of the fed back position is as follows:

0 000010 002000 FFFFFE
A B C

A Position x, hexadecimally, using a twin complement, in the example, the decimal value 16.
B Position y, hexadecimally, using a twin complement, in the example, the decimal value 8,096.
C Position z, hexadecimally, using a twin complement, in the example, the decimal value 2.

Restrictions This command can only be used if no traversing movement is carried out (if the plant is stopped).
The command cannot be transmitted during the execution of stored commands.
The processor card cannot check whether the set position of the current position corresponds to the current position of the mechanical system, since no closed-loop control circuit exists.

Programming example

PAL-PC (terminal mode)	GW-BASIC
@0P	-

In any case, the positions of all three axes are fed back by the function, irrespectively of the number of axes defined.
The interface card will send the ASCII characters at the set transmission rates without expecting a confirmation from the receiving computer using the hardware handshake.

2.1.7 Command: Zero offset

Application

The processor card will store the current position as a virtual zero point for the specified axis (axes).
The next commands of the type *Absolute Movement* will use this virtual zero point as the new reference point.

Structure

@<GN>n<axes>

<GN> = device number, default = 0

<axes> = integer value between 1 and 7

Notation

@0n7 @0n1

. .

Explanation

The card is addressed using @0. "n" specifies that a zero offset is to be carried out. After this command, the computer will receive the information for which axes a zero offset is to be carried out.
The assignment will be x = 1, y = 2, z = 4.
If a zero offset is to be carried out for several axes, the above values must be added:

Axes	Value	Axes	Value
x	1	xy	5
y	2	yz	6
z	4	xyz	7
xy		3	

After the command has been carried out, the computer will provide a check-back signal (cf Software Handshake).

Restrictions

The virtual zero point is only relevant for the command *Absolute Movement*. The positioning using incremental dimensions will not be affected by the virtual zero point, since a traversing vector is specified here.

Programming example

PAL-PC

```
#axis xy;
#elev 4,4;
moveto 80(900),8(900);
null xy;
moveto 2(900),4(990);
stop.
```

GW-BASIC

```
  100 open“com1:9600,N,8,1,DS,CD“as #1
  110 print#1,“@03":gosub 1000
  120 print#1,“@0r3":gosub 1000
  130 print#1,“@A1000,2000,2000,2000":gosub 1000
  140 print#1,“@0n3":gosub 1000
  150 print#1,“@M100,2000,100,2000":gosub 1000
  160 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$=“0" then return
1020 print ”card signals error: ”;a$
1030 stop
```

Referencing will relocate the virtual zero point to the plant zero point.

2.1.8 Command: Select plane

Application 2.5D interpolating processor cards (e.g. Interface Card 4.0) can interpolate only two of three axes. These are the X and Y axes (provided they are turned on). The *Select Plane* command, however, can be used to define any plane configuration other than the main plane. The remaining third axis will be considered as the tool axis and will then be traversed, i.e. after positioning of the main axes.

Structure @<GN>e<plane>

<GN>	=	device number, default = 0
<plane>	=	a number between 0 and 2
0	=	xy
1	=	xz
2	=	yz

Notation

@0e1 Switch to xz interpolation
@0e0 Switch to xy interpolation

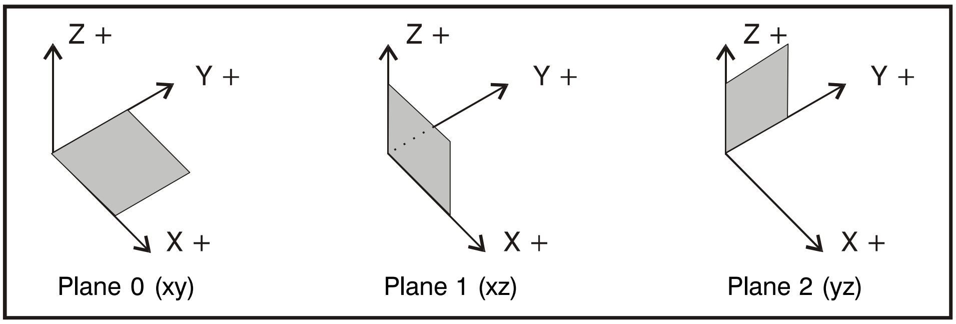


Fig. 2: Switch-selectable interpolation planes

Explanation

To achieve high speeds (a maximum axis speed of 10 kHz corresponds to a vector speed of 175 mm/s in half-step mode at a leadscrew pitch of 5 mm and a vector of 45°), the processor card can calculate only the speed ratios of two axes another to one within this time. The Plane command can be used to switch between the interpolation planes without loss in speed.

Restrictions

If an interpolation plane other than XY is selected, zero should be transferred as the number of steps for the second movement of the machining axis.

Programming example

PAL-PC

```
#axis xyz;
line yz;
move 20(1000),30(1000),
        33(1000),0(30);
stop.
```

GW-BASIC

```
 100 open“com1.9600,N,8,1,DS,CD“as #1
 110 print#1,“@07":gosub 1000
 120 print#1,“@0e2”
 125 gosub 1000
 130 print#1,“@0M20,200,30,900,33,900, 0,21”
 135 gosub 1000
 140 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$=“0" then return
1020 print ”card signals error: ”;a$
1030 stop
```

In the example above, the Y and Z axes are interpolated (traversed to the target position along a straight line), and the X axis follows up.
The plane selection has no influence on the referencing order.
If you wish to modify the referencing order, do not transfer referencing commands that contain axis combinations.

2.1.9 Command: Peek (read memory address)

Application The *Peek* command can be used to poll the contents of a memory cell of the processor card both in the data memory and in the read-only memory via the serial interface.

Structure
@<GN>c<Addr> (read-only memory)
@<GN>b<Addr> (random-access memory)

<GN> = device number, default = 0
<Addr> = address between 0 and 65,536

Notation
@0c 2048
@0b 4711

Explanation The card is addressed using @0. “c“ specifies that a value is to be read from the read-only memory. “2048“ specifies the address of the value to be read. The computer will reply with the software handshake followed by two characters that specify a hexadecimal value corresponding to the contents of the memory cell. To read a value from the data memory, use command code “b“ instead of “c“.

Restrictions -

Programming example

PAL-PC (terminal mode)	GW-BASIC
@0b 65531	-

This command is used in its extended form in conjunction with an I/O expansion unit (see Section 2.3.3, Commands that can be stored: *Read input port*).

2.1.10 Command: Poke (write memory address)

Application The *Poke* command can be used to modify the contents of the data memory of a processor card.

Structure @<GN>B<Addr>,<Data>

<GN> = device number, default = 0
<Addr> = address between 0 and 65,535
<Data> = value between 0 and 255

Notation @0B 33211,128

Explanation The card is addressed using @0. “B“ specifies that a value is to be written into the memory. “33211“ specifies the address of the value to be written. “128“ is the new value of this memory cell.
The computer will confirm the execution of the command with the software handshake.

Restrictions The command will not check whether a device connected to the data bus has accepted the data correctly.

Programming example

PAL-PC (terminal mode)	GW-BASIC
@0B33211,128	-

This command should not be used to modify internal card parameters, since the card address may change without prior notice. The command should not be used with addresses less than 32767, since these addresses are used by the processor card as the data memory.

2.1.11 Command: Clear battery-backed RAM

Application This command is used to delete all RAM data even if they are stored quasi-continuously thanks to the *Memory Back-up* option. This operation will also reset any information on reference speed, axes etc.

Structure @<GN>k (directly)

<GN> = device number, default = 0

Notation @Ok

Explanation The card is addressed using @0. “k“ specifies that the battery-backed RAM is to be cleared. After this command has been executed, the computer will provide an appropriate check-back signal (cf Software Handshake).

Programming example

PAL-PC (terminal mode)	GW-BASIC
@0k	100 open“com1:9600,N,8,1,DS,CD“as #1
	110 print#1,“@Ok“:gosub 1000
	120 stop
	1000 if loc(1)<1 then goto 1000
	1010 a$=input$(1,1)
	1015 if a$=“0" then return
	1020 print ”card signals error: ”;a$
	1030 stop

If the RAM can no longer be cleared in this way, remove it from its socket for a short moment and then reinsert it.

2.1.12 Command: Set CR/LF

Application When receiving data, some computers must be provided with a delimiter character at the serial interface. This is an operating system-specific feature of some process computers (e.g., DEC VAX, HP process computers).
The process computer needs the delimiter character to be able to initiate an interrupt for the received process at the end of the data transfer.

If you are using an MS-DOS computer (IBM-PC, XT, AT or the like), this command should not be used.

Structure @<GN>C<STATUS> (directly)

<GN> = device number, default = 0
<STATUS> = 0 = OFF (default), 1 = ON

Notation @0C1

Explanation The card is addressed using @0. “C“ specifies that the software handshake is to be modified. In this case, with all commands, the card will check back with the sequence:

Error CHR(13) CHR(10)

Restrictions The card can then only be addressed using the new protocol.
To switch to the new protocol, use the new protocol for transfer.

A programming example cannot be specified here, since neither PAL-PC, nor GW-BASIC is available on such process computers.

2.1.13 Command: Set device number

Application This command is used to modify the device number of the process card (<GN>). Numbers between 0 and 9 (chr(48) ... chr(58)) are admissible. The new device number remains active until the device is switched off.

Structure @<GN>G<GNnew> (directly)

<GN> = device number, default = 0
<GNneu> = character between 0 and 9

Notation @0G1

Explanation The card is addressed using @0. “G“ specifies that a new device number is to be used to address the card.
After “G“, the card will expect the new device number.
After storing, the computer will provide an appropriate check-back signal (cf Software Handshake).

Restrictions The card can then only be addressed using this device number. The card will not check whether an admissible device number is transferred.

Programming example

PAL-PC (terminal mode)

@0G1

GW-BASIC

```
 100 open“com1:9600,N,8,1,DS,CD“as #1
 110 print#1, ”@03":gosub 1000
 120 print#1,”@0G1":gosub 1000
 130 print#1,“@1i“:gosub 1000
 140 print#1,“m 8000,900,800,900":gosub 1000
 150 print#1,“n 3":gosub 1000
 160 print#1,“m 200,900,400,990":gosub 1000
 170 print#1,“9":gosub 1000
 180 print#1,“@1S“:gosub 1000
 190 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$=“0" then return
1020 print ”card signals error: ”;a$
1030 stop
```

To address a processor card with a modified device number in PAL-PC, use the command #GN. In this case, the processor card can only be addressed using this device number.

2.1.14 Command: TRACE (single-step mode)

Application This command will cause the processor card to execute the stored commands separately. After each command, the processor card waits for a character at the serial interface, and the command count is output with all relevant parameters.

Structure @<GN>t

<GN> = device number, default = 0

Notation @0t

Explanation The processor card will carry out the commands as usual; prior to each command, however, the command count is output as an integer number. After the command count, the command number and the operation constant with the relevant data is output. The line is completed with CR. The computer will then wait for a character at the interface. Then the command is executed. With each command to be executed, the function will behave as follows:

(A): The trace string is output (see below).
The card is waiting for a character.
If character = 127 is provided, the microprocessor will be reset.
The command will be carried out.
The command = end of data field will complete the process.
Otherwise, the next cmmand wil be provided after (A).

The trace string transferred with each command has following structure:

Structure

01234	00001	30	000001	FE87 ...	FFFF01	FE01
A	B	C	D	E	F	G

A Memory pointer - specifies where the command is stored in the memory of the processor card.

B Command counter - specifies the number of the current NC command.

C NC command code - specifies the command to be output. The command is specified hexadecimally with reference to the ASCII value of the corresponding command code. In the example above, the command 0 = relative movement is stored.

D Command parameter of X axis; in the example above, the representation of the distance to be traversed is a 24-bit hexadecimal number using the twin complement mode of representation.

E Speed value of X axis; to reconvert the speed, the fraction 921600/(high byte*(256-low byte) can be generated.

F Command parameter as D, but for the z2 specification.

G Speed specification for the z2 specification.

As with other commands, the parameters will be stored in the transferred order either as a character or as a twin complement.

With synchronisation commands, first the character for the trace function must be transferred and then the synchronisation character. To switch the execution of the individual steps, do not use the characters <spacebar>, <TAB> and <linefeed>.
To exit Trace mode, either switch off the processor card or transfer DEL (char(127)) if a character is requested.
In the individual software versions, the order and mode of saving of the parameters is subject to changes within the framework of the technical progress without prior notice. The number of command parameters sent back corresponds to the number of axes selected.

2.1.15 Command: Self-test

Application This command is used to initiate a self-test of the processor card. In contrast to the self-test initiated by actuating the start button, this command is only used to carry out the first part of the subroutine, and not the traversing and interface test.

Structure @<GN>? (directly)

<GN> = device number, default = 0

Notation @0?

Explanation The card is addressed using @0. “?“ specifies that a self-test is to be carried out. The card will then test the memory area, processor and processor register, as well as the internal memory areas. Then some processor card-related variables and a checksum are output.
To carry out the expanded self test, hold down the Start button and turn on the device.

To exit the self-test, first turn off the device; no other commands can be transferred unless the self-test is completed!

Programming example

PAL-PC (terminal mode)	GW-BASIC
@0?	-

To be able to transfer further commands to the processor card, first the self-test must be completed. Otherwise, a list character (error @) will be sent as an error message. In this case, error 164 will be signalled to the PAL-PC.

2.2 Supplementary Command Set of Interface Card 5.0

2.2.1 Command: 3D linear interpolation

Application

Interface Card 5.0 expands the 2.5D interpolation of the standard operating system to a 3D interpolation.
You can use this command to enable/disable the interpolation as required for the particular task in question.

Structure

@<GN>z<STATUS>

<GN>	= device number, default = 0
<STATUS>	= 0 - 3D interpolation OFF
	= 1 - 3D interpolation ON

Notation

@0z1

Explanation

The data opening part @0 will prepare the processor card for a new command. “z1“ will modify the interpolation from 2-axis to 3-axis operation.

The instruction has a modal effect, i.e. all MOVE and MOVETO instructions are carried out as 3D statements. The specification of z2 parameters will be ignored in these traversing movements.
The specification of the speed for the interpolation must be performed with the X specification.

Programming example

<u>PAL-PC</u>

```
#axis xyz;
reference xyz;
set3don;
move 10(700),15(800),3(400),
        0(30);
set3doff;
```

<u>GW-BASIC</u>

```
100 open“com1:9600,N,8,1,DS,CD“as #1
110 print#1,“@07":gosub 1000
120 print#1,“@0r7":gosub1000
130 print#1,“@0z1":gosub 1000
140 print#1,“@0A100,700,150,800,30,400,0,30"
145 gosub 1000
150 print#@1,“@0z0":gosub1000
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$=“0" then return
1020 print ”card signals error: ”;a$
1030 stop
```

Referencing will switch back the system automatically to 2.5D interpolation.
The correct processing of a 3D interpolation requires an XY plane as the reference plane (cf Plane Selection).

The maximum speed for a 3D interpolation is 10,000 steps/s.

The speed that can be achieved by the mechanical system depends on the connected motors and power sections.

To carry out movements at rapid traverse, you should switch to 2D interpolation for a short time and carry out positioning with the Z axis lifted, since no collision check is carried out during a 3D interpolation.

In case of a 3D interpolation, the current position will not be correct after a Stop command.
The position interrogation will provide the correct values only after the command has been executed successfully.

2.2.2 Command: Circular interpolation

Application This command is used for processing circles and arcs at constant traversing rate. The circular interpolation is initiated by two successive commands. The first command defines the circle direction, and the second one is used to transfer the interpolation parameters.

Structure Circle direction: @<GN>f-1 CCW
@<GN>f0 CW

Arc: @<GN>y B,V,D,Xs,Ys,Rx,Ry

B	= arc length in steps
V	= speed (30 ... 10,000)
D	= interpolation parameter
Xs	= start point x
Ys	= start point y
Rx	= direction x
Ry	= direction y

Calculating the parameters

Arc length B The arc length specifies the length of the arc between the starting and the end points of the interpolation in steps. To calculate this parameter, you can also use the program parts listed below. The following rule applies:

A - starting angle of arc or circle segment
A = pi*starting angle/180

E - end position of movement
E = pi*end angle/180

B - resulting arc length

To calculate the arc length, you may only use angles specified in arc dimension.

1. Approximating formula (only with quarter, semi and full circles)
B = 4*radius*(E-A)/pi

2. Calculating the arc length using a software routine

```
if (circle direction=CCW) then
  begin
        while(A<0) do A:=A+2.0*pi;
        while(E<0) do E:=E+2.0*pi;   {scale angle to positive range}
        while (A>=pi/2.0) do
          begin
                A:=A-pi/2;
                E:=E-pi/2;
          end;
        B:=0.0;
        while (E-A>=pi/2.0) do
          begin
                E:=E-pi/2.0;
                B:=B+2.0*radius;
          end;
        B:=B+raduis*(cos (A) -cos (E) +sin (E) -sin (A));
  end;
else    {circle direction = CW}
  begin
        while (A>0) do A:=A-2.0*pi;
        while (E>0) do E:=E-2.0*pi;   {scale angle to positive range}
        while (A<=-pi/2.0) do
          begin
                A:=A+pi/2;
                E:=E+pi/2;
          end;
        B:=0.0;
        while    (A-E>=pi/2.0) do
          begin
                E:=E+pi/2.0;
                B:=B+2.0*radius;
          end;
        B:=B+radius*(cos (A) -cos (E) +sin (A) -sin (E));
   end;
if (B<0) then B:= -B;
```

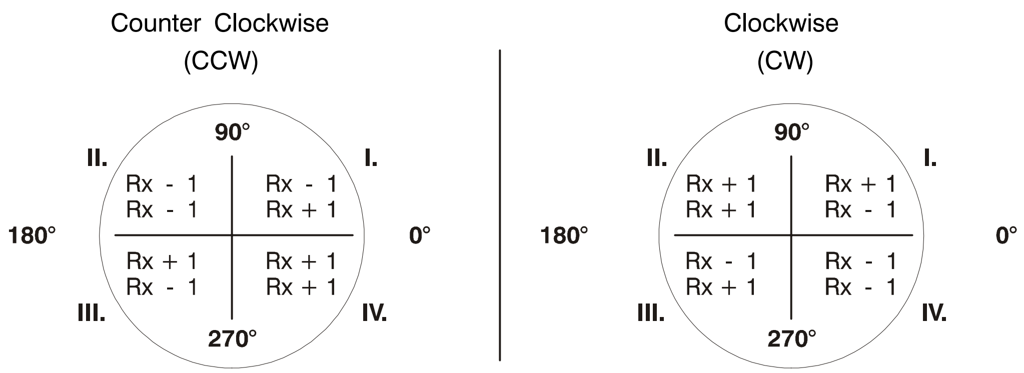
The calculated arc length must be transferred to the next, integer value as a rounded value. Values ranging from 3 to 8,000,000 are admissible in steps.

The velocity V

Integer values ranging from 30 to 10,000 steps/s are admissible for the velocity. Whether and at which velocities the interpolation can be carried out, depends on the power sections used and the mechanical system connected.

The directions Rx and Ry

The parameters Rx and Ry are used to tell the processor card in which quadrant of the circle the processor card the interpolation will start.



The starting points Xs and Ys

These parameters specify the starting points Xs and Ys relative to the circle centre. The following formulas are used for calculating:

Xs = radius * cos(A)

Ys = radius * sin(A)

The interpolation parameter D

“D“ has to be transferred, since the processor card is not able to calculate this parameter due to its memory capacity.
For calculation, you can use the program section below:

```
function  sum(xx:real):real;
  begin
     if(xx>0)  then
        sum:=xx*(xx+1)
     else
        sume:= -xx*(xx-1)
  end;

function  formel:real;
  begin
     if  (circle  direction=CCW)  then
        formula:=   (Rx*Ry*radius+Rx*Ry*sum(radius-1.0)
                     -Rx*sum(Xs+(Rx-Ry)/2.0)+Ry*sum(Ys+(Rx+Ry)/2.0))/2;
     else                      {direction  =  CW}
        formula:=   (-Rx*Ry*radius-Rx*Ry*sum(radius-1.0)
                    -  Rx*sum(Xs+(Rx+Ry)/2.0)  +  Ry*sum(Ys+(Ry-Rx)/2))/2;
  end;
D:=formula;
```

☞ The calculated parameter must be transferred as a rounded and integer value.

Programming example for calculating the parameter:

A quarter arc CCW circle (90°) having a radius of 200 steps and a traversing rate of 1,500 steps/s is to be carried out. The starting angle is 135°.

1. Angle specified in arc dimension:

   A = pi*135/180
     = 3*pi /4
   B = pi* (135+90) /180
     = 5*pi /4

2. Arc length: (using the approximation formula)

   B = 4*radius*(E-A) /180
     = 2*200*(5*pi /4-4*pi) /pi
     = 400

3. Directions Rx, Ry:

   Starting angle= 135°, CCW
   Rx = - 1
   Ry = - 1

4. Starting positions Xs, Ys:

   Xs = radius*cos(starting angle)
     = 200*cos(135)
     = -141
   Ys = radius*sin(starting angle)
     = 200*sin(135)
     = 141

5. Interpolation parameter D: (CCW)

D = (Rx*Ry*radius + Rx*Ry*sum(radius -1.0)
- Rx*sum(Xs+(Rx-Ry) /2.0) + Ry*sum(Ys+(Rx+Ry) /2.0)) /2

Sum(radius-1)	=	sum (200-1)
	=	sum (199)
	=	199*(199+1)
	=	39 800
Sum(Xs+(Rx-Ry) /2.0)		
	=	sum (-141 + (-1 -(-1)) /2.0)
	=	sum (-141)
	=	-(-141)*((-141) -1)
	=	- 20 022
Sum (Ys+(Rx+Ry) /2.0)	=	sum (141 +(-1 +(-1)) /2.0)
	=	sum (141-1)
	=	140*(140+1)
	=	19 740

D	=	(Rx * Ry*radius + Rx* Ry*39800 - Rx*(-20022) + Ry*19740) /2
	=	((-1)*(-1)*200 + (-1)*(-1)*39800 - (-1)*(-20022) + (-1)*19740) /2
	=	(200 + 39800 - 20022 - 19740) /2
	=	119

The program section must look as follows:

```
...
...
@0f-1
@0y400,1500,119,-141,141,-1,-1
...
...
```

or, in the direct format:

```
...
...
f-1
y400,1500,119,-141,141,-1,-1
...
...
9
@0s
```

2.3 Supplementary Command Set of Interface Cards with I/O Expansion

2.3.1 Command: Save externally

Application This command is used to save a CNC program on an external storage medium.

Structure @<GN>u
<GN> = device number, default = 0

Explanation This command can be used to save a CNC program currently stored in the data memory of the processor card on a data memory in check card format (memory card) or to read it back from a memory card to the processor card.
When doing so, observe the following sequence:

1. Transfer from processor card to memory card
   a) Transfer your program to the processor card as usual.
   b) Insert the memory card.
   c) Transfer the @0u command.
   d) Remove the memory card.

2. Transfer from memory card to processor card
   a) Turn on the controller.
   b) Insert the memory card.
   c) Push µP Reset.
   d) Remove the memory card.

☞ When turning on the control system, the memory card should not be installed. With the memory card types 8 k x 8 and 16 k x 8, the existing memory capacity will not be checked, i.e. in case of complex programs, the memory limits may be exceeded without error message.

2.3.2 Command: Set output port

Application The processor card will set a defined output pattern at the defined output port of the I/O unit.

Structure @<GN>B<ADDRESS>, <VALUE>

<GN>	= device number, default = 0
<ADDRESS>	= output port 1 ——> 65 529
	= output port 2 ——> 65 530
<VALUE>	= 0 ... 255

Explanation This command corresponds to a large degree to the Poke command of the standard operating system 5.x. During CNC operation (memory mode), in addition to byte-by-byte processing, bit-by-bit processing of the output ports is possible, allowing you to set or delete individual bits separately.

2.3.3 Command: Read input port

Application The processor card will read in the bit pattern provided at the input port of the I/O expansion.

Structure @<GN>b<ADDRESS>

<GN>	= device number, default = 0
<ADDRESS>	= input port 65 531

Explanation The structure of this command is identically to that of the *Peek* command used in the standard operating system 5.x. In addition, in memory mode (CNC operation), it is possible to carry out backward and forward branches, depending on the bit pattern.

2.4 Supplementary Command Set of EP1090

2.4.1 Command: Output module

Application This command is used to connect/disconnect the main spindle, i.e. the interface card will switch on/switch off the main spindle.

Structure

@<GN>h<parameter>	(DNC)
h<parameter>	(can be stored)
<GN>	= device number, default =0
<Parameter>	= 1 Main spindle ON
	= 0 Main spindle OFF

Notation @0h1

Explanation The card is addressed using @0. “h“ specifies that the output module is to be switched. Parameter “1“ will switch on the integrated relay. If the spindle is already in the ON condition, the command will have no effect. The current condition of the main spindle is indicated by the decimal point of the 7-segment display.
After the command has been executed, the computer will check back “0“.

2.5 Supplementary Command Set for Interface Card, Version AZ1350/5 and Higher

2.5.1 Command: Magnetic brake

Application This command can be used for software-controlled enabling/disabling of a special output that controls magnetic brakes in drive units.

Structure @<GN> g <status>

<GN>	= device number
<Status>	= 1 Brake magnetised
	= 0 Brake inactive

Notation @0g1

Explanation The card is addressed using @0, “g“ specifies that the brake relay is to be switched. Status “1“ will turn on the relay. This will activate a connected magnetic brake and will release the driving axis of the motor. In 0 condition, the brake will not be supplied with current, thus braking the driving axis of the motor.
After this command has been carried out, the computer will check back “0“.

2.6 Check and Control Codes

Check and control codes provide direct access to the sequence of functions of the interface card via the serial interface. The transmitted commands are carried out without delay.

2.6.1 Command: Self-test

Application The processor card will test the operational performance of its function modules.

Structure chr(252)

Explanation The interface card will check the capacity of its data memory, the checksum of its operating EPROM and the switch position of the DIP switch. Then, for testing the connected stepper motors, some clock pulses are output to the power electronics of the X and Y axes. The test routine is completed by a permanent output of an ASCII character set via the serial interface.

Restrictions -

Programming example

PAL-PC (terminal mode)	GW-BASIC
chr(252)	-

You can complete the self-test only if you turn off the supply voltage or carry out a μP reset.

2.6.2 Command: STOP

Application The processor card will interrupt the current traversing movement.

Structure chr(253)

Explanation In DNC mode, a positioning movement (relative or absolute) can be interrupted by a STOP command without step losses. A START pulse executed after the STOP command will complete the interrupted sequence of functions. Furthermore, you can read back the currently reached position after a STOP command using the *Position Interrogation* command.

Restrictions You can use the command only if a positioning movement is carried out.

Programming example

PAL-PC (terminal mode)	GW-BASIC
chr(253)	-

The processor card will feed back the stop error as the acknowledgement signal.

Since the command operates without addressing, the traversing movements of all connected processor cards will be interrupted.

The higher-level computer must retransmit the position to be approached last in DNC mode.

2.6.3 Command: µP Reset

Application The processor card will abort all activities immediately and will change to the RESET state.

Structure chr(254)

Explanation A µP reset will switch back the integrated microcontroller to its initial state without delay. During the reset state, the outputs have Vcc potential and will switch off when the GND potential is no longer present.

Restrictions -

Programming example

<u>PAL-PC</u> (terminal mode)	<u>GW-BASIC</u>
chr(254)	-

2.6.4 Command: Break

Application The Break command is used to cancel the current positioning process.

Structure chr(255)

Explanation Sending off a Break command will cancel the current positioning process of the interface card without initiating a stop ramp. Any resulting step errors will be ignored. In contrast to the µP reset command that has a similar effect, you can go on working after the *Break* command as usual without reinitialising the processor card.

Restrictions -

Programming example

<u>PAL-PC</u> (terminal mode)	<u>GW-BASIC</u>
chr(255)	-

3 CNC Command Structure

In CNC mode, the processor card stores all transmitted commands in the internal data memory. To activate them, the command @<device number>i has to be transferred after the standard initialisation with @<device number> <number of axes>. Then the data file is transferred and completed using command 9 as the end-of-data-field character. Now you can reactivate the program using an external Start command.
For starting, you can use both a Start button (e.g., on the front panel of the processor card) and the command @<device number>S.

Due to the physically limited RAM of the processor cards, the number of commands that can be stored is limited to approx. 1,200 in 3-axis mode, approx. 1,800 in 2-axis mode and approx. 2,400 in 1-axis mode.

In order to avoid data loss in the RAM in case of failure of the supply voltage (e.g., when the supply voltage is switched off), a so-called memory backup can be provided by installing an accumulator or a primary cell available as options.

The commands that can be stored are listed and explained in brief below. For a detailed explanation, please refer to direct mode of the corresponding command.

3.1 Basic Command Set of Processor Card 4.0 and Higher

3.1.1 Command: INPUT

Application This command will set the processor card to the memory mode. All next following commands will be stored in the internal data memory. The stored commands can be executed using either the Start command or pressing the Start button.

Structure @<GN>i
<GN> = device number, default = 0

Notation @0i

Explanation The card is initialised using @0. “i“ specifies that commands follow, which are to be stored. After the command has been received, the processor card expects a complete NC program consisting of commands that can be stored. This program has to be completed using an end-of-data-field character (9).
The data field may contain all commands that can be stored.

Restrictions You can use the command only after the number of axes has been set, i.e. the processor card has been initialised.

Programming example

PAL-PC

```
#axis xy;
move 50(100),40(100);
stop.
#start
```

GW-BASIC

```
 100 open“com1:9600,N,8,1,DS,CD“as #1
 110 print#1,“@03":gosub 1000
 120 print#1,“@0i“:gosub 1000
 130 print#1,“0 50,100,40,100":gosub 1000
 140 print#1,“9":gosub 1000
 150 print#1,“@0S“:gosub 1000
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$=“0" then return
1020 print ”card signals error:”;a$
1030 stop
```

The *Input* command will delete all statements stored to date. The processor card will then expect a complete data field, comleted with the end-of-data-field character (9).

If an error occurs during the transfer of the commands to be transferred, the processor card will quit the Input mode, and the NC program transferred to date get lost.

3.1.2 Command: Reference Point Approach

Application The processor card will traverse all specified axes to their zero points (reference points).

Structure 7<axes>

<axes> = integer value between 1 and 7

Explanation “7“ specifies that reference point approach is to be carried out. The following numerical value defines all axes to be referenced.

x	=	1	xz	=	5
y	=	2	yz	=	6
xy	=	3	xyz	=	7
z	=	4			

The order of execution is defined as follows:

—> Z axis —> Y axis —> X axis

This is also true if an axis other than the tool axis has been defined using the Plane command. As required, approaching of the individual axes to their reference points separately may avoid collisions with the workpiece.

Programming example

<u>PAL-PC</u>

```
#axis xyz;
reference xyz;
```

<u>GW-BASIC</u>

```
 100 open“com1:9600,N,8,1,DS,CD“as #1
 110 print#1,“@07":gosub 1000stop.
 120 print#1,“@0i“:gosub 1000
 130 print#1,“77":gosub 1000
 140 print#1,“9":gosub 1000
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$=“0" then return
1020 print ”card signals error:”;a$
1030 stop
```

(see *Reference Point Approach* command, Section 2.1.2)

3.1.3 Command: Relative Movement

Application According to the transferred number of steps and the step velocity, the processor card provides a certain step sequence for each power output stage.

The traversing movement will either be carried out immediately or stored.

Structure 0 <Sx>,<Gx>,<Sy>,<Gy>,<Sz1>,<Gz1>,<Sz2>,<Gz2>

0 = relative movement
<Sx> = number of steps x, value between 0 and ± 8,388,607
<Gx> = veloxity x, value between 30 and 10,000
.
.
<Gz2> = velocity of Z axis (2nd movement)

Explanation “0“ specifies that a relative movement is to be carried out.
The processor card will now expect a pair of numbers consisting of the number of steps and the speed for each axis.

Programming example

PAL-PC

```
#axis xyz;
move 50(500),300(900),
     20(200),-20(900);
move 20(300),300(3000),
     0(21),0(21);
stop.
```

GW-BASIC

```
 100 open“com1:9600,N,8,1,DS,CD“as #1
 110 print#1,“@03":gosub 1000
 120 print#1,“@0i“:gosub 1000
 130 print#1,“0 35,800,250,2000":gosub 1000
 140 print#1,“0 20,2000,-25,1000":gosub 1000
 150 print#1,“9":gosub 1000
 160 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$=“0" then return
1020 print ”card signals error: ”;a$
1030 stop
```

(see *Relative Movement* command, Section 2.1.4)

3.1.4 Command: MoveTo (position)

Application The processor card will traverse to the specified position at the specified traversing rates. The traversing movement will be carried out immediately.

Structure m <Sx>, <Gx>, <Sy>, <Gy>, <Sz1>, <Gz1>, <Sz2>, <Gz2>

Explanation “m“ specifies that an absolute position will follow.

For reasons of compatibility with the relative movement command, two pairs of numbers are expected for the Z axis also in this case. The second value of the Z position must be zero. Although this number will be ignored, it must be specified.

Programming example

<u>PAL-PC</u>

```
#axis xy;
moveto 50(500),300(900);
moveto 50(500),300(900);
moveto 20(200),30(900);
stop.
```

<u>GW-BASIC</u>

```
 100 open“com1.9600,N,8,1,DS,CD“as #1
 110 print#1,“@03":gosub 1000
 120 print#1,“@0i“:gosub 1000
 130 print#1,“m 500,800,200,31":gosub 1000
 140 print#1,“m31,500,40,500":gosub 1000
 150 print#1,“9":gosub 1000
 160 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$=“0" then return
1020 print ”card signals error: ”;a$
1030 stop
```

(see Befehl *MoveTo* command, Section 2.1.5)

3.1.5 Command: Zero offset

Application The processor card will store the current position as the virtual zero point for the specified axis (axes).
The next commands of the type *Traverse using absolute dimensions* will take into account this virtual zero point as the new reference point.

Structure n<axes>

<axes> = integer value between 1 and 7

Explanation "n" specifies that a zero offset is required. After this command has been executed, you must tell your computer for which axes you wish to carry out a zero offset. The assignment is x = 1, y = 2, z = 4. If you wish to carry out a zero offset for several axes, the values above must be added.

Programming example

PAL-PC

```
#axis xy:
move 350(800),200(800);
null xy;
move 20(500),30(300);
stop.
```

GW-BASIC

```
 100 open"com1:9600,N,8,1,DS,CD"as #1
 110 print#1,"@03":gosub 1000
 120 print#1,"@0i":gosub 1000
 130 print#1,"0 8000,900,800,900":gosub 1000
 140 print#1,"n 3":gosub 1000
 150 print#1,"m 200,900,400,990":gosub 1000
 160 print#1,"9":gosub 1000
 170 print#1,"@0S":gosub 1000
 180 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$="0" then return
1020 print "card signals error: ";a$
1030 stop
```

(see *Zero Offset* command, Section 2.1.7)

3.1.6 Command: Select plane

Application

2.5-D interpolating processor cards (e.g., Interface Card 4.0) can interpolate only two of three axes. In the ON condition, this pertains to the X and Y axes. The *Select Plane* command can be used to define any plane configuration other than the main plane. The remaining third axis will be considered as the tool axis and be traversed to the positions of the main axes.

Structure

e<plane>

<plane>= number between 0 and 2

0 = xy
1 = xz
2 = yz

Explanation

(see Section 2.1.8)

Programming example

<u>PAL-PC</u>

```
#axis xyz;
line yz;
move 20(200),33(500),
        40(1000),0(21);
stop.
```

<u>GW-BASIC</u>

```
 100 open“com1.9600,N,8,1,DS,CD“as #1
 110 print#1,“@07":gosub 1000
 120 print#1,“@0i“:gosub 1000
 130 print#1,“e2":gosub 1000
 140 print#1,“m20,200,30,900,33,900, 0,21"
 150 gosub 1000
 160 print#1,“9":gosub 1000
 170 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$=“0" then return
1020 print ”card signals error: ”;a$
1030 stop
```

(see *Select Plane* command, Section 2.1.8)

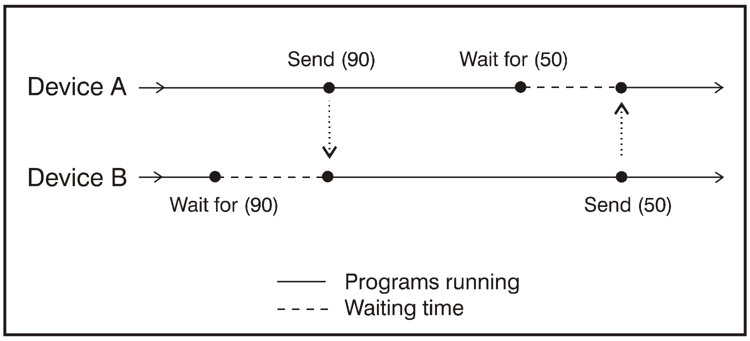
3.1.7 Command: Transmit synchronisation character

Application The processor card will tell a second processor card or a higher-level computer that a certain point in the sequencing diagram (NC program) is reached. This command is used to synchronise the processor card with an external unit or to request an external unit to do an activity.

Structure 1 <SyncChar>

<SyncChar> = synchronisation character between 33 and 125

Explanation The processor card will send a defined ASCII character to the serial interface. Due to the *Wait for Synchronisation Character* command, the receive station will wait for the appropriate character and will continue with the programmed CNC sequence after the character has been received. The Diagram below provides a short overview of the sequence of functions.



Restrictions Due to the commands *Send Synchronisation Character* and *Wait for Synchronisation Character,* only two processor cards can be synchronised without higher-level computer.
The transferred number of the synchronisation character must be a printable character in the range between 33 and 125, since other characters are filtered by the processor card. The character "64" should not be used, since this character will open the data traffic of waiting processor cards. The serial interfaces of the devices must be linked using the interface/interface link cable.

Programming example

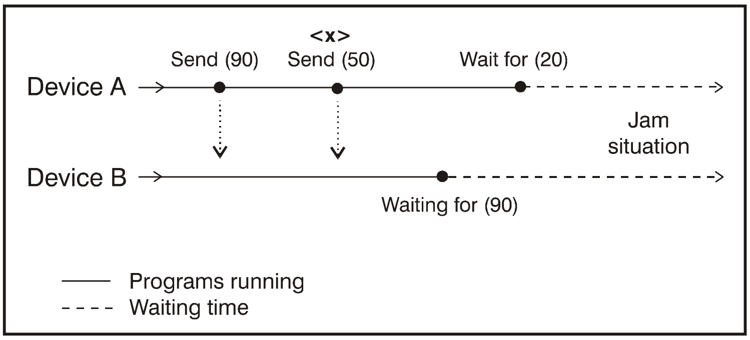
PAL-PC

```
#axis xyz;
#input
send 90;
.
.
.
```

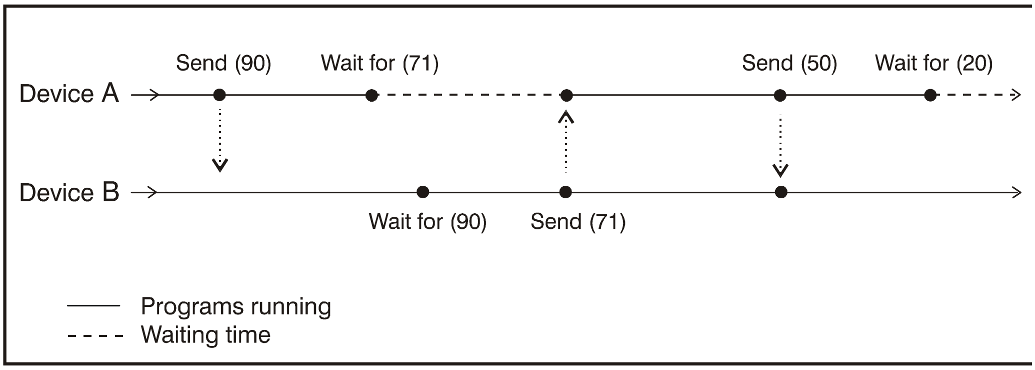
GW-BASIC

```
  100 open"com1:9600,N,8,1,DS,CD"as #1
  110 print#1,"@07":gosub 1000
  120 print#1,"@0i":gosub 1000
  130 print#1,"1 90":gosub 1000
  140 print#1,"9":gosub 1000
  150 print#1,"@0s":gosub 1000
  160 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$="0" then return
1020 print "card signals error: ";a$
1030 stop
```

To test the PAL-PC program, you can use the *Communication* function, and to test the BASIC program, you can use the program used for the interface test.
While a stored command sequence is executed, the interface card can receive and buffer only one character. The following situation will therefore necessarily result in a total jam of the entire system:



With <x>, a character char(50) is contained in the input buffer of device "B" (a character sent before this character has been overwritten). The process will thus wait "for ever" for the required character char(90). For this reason, the sending device should wait for a confirmation of the receiving device before a new synchronisation character is sent.



3.1.8 Command: Wait for synchronisation character

Application The processor card waits for the reception of the specified character at the serial interface. In conjunction with a higher-level computer, this command can be used for branches within the stored sequence.

Structure 2 <SyncChar>, <offset>

<SyncChar> = synchronisation character from 33 to 125

<Offset> = branch when receiving <SyncChar>+1
number between - 32,767 and + 32,767

Notation

250.0	Waiting for the synchronisation character
50,255.7	Waiting for 55, branch forward by 7 commands when receiving 56

Explanation For the use of the command, please refer to ”Send Synchronisation Character”.
In conjunction with a higher-level computer, you can use this command for logical decisions within the process sequence:

Program step

1
2
3 wait 50,-1 <— ext. computer sends 50 or 51
4
5

The data field will stop at command “3“. If the higher-level computer sends char(50), No. 4 will be executed as the next command; if the computer sends char(51), command No. 2 will be executed as the next command.
Generally: If the processor card receives the character following after the character for which the processor card is waiting, the specified branch will take place; otherwise, the command following after the waiting command will be carried out.

Programming example

PAL-PC

```
#axis x;
input
label: move 3(1000);
wait 50,label;
stop.
#start
```

GW-BASIC

```
 100 open"com1:9600,N,8,1,DS,CD"as #1#
 110 print#1,"@01":gosub 1000
 120 print#1,"@0i":gosub 1000
 130 print#1,"0 500,5000":gosub 1000
 140 print#1,"2 50,-1":gosub 1000
 150 print#1,"9":gosub 1000
 160 print#1,"@0s":gosub 1000
 170 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$="0" then return
1020 print "card signals error: ";a$
1030 stop
```

After the transfer to the processor card has been carried out, this program will be started first with @0s. The next step in the program sequence is a relative movement of the X axis. Then the processor card expects char(50) or char(51) at the serial interface. If char(51) has been transmitted, the card will branch back, and the relative movement is carried out again. If char(50) is received, the program is quitted.

Please note that branching to a position before or after the end of the data field may produce unforeseable results.

3.1.9 Command: Loop / Branch

Application Program loops are used to summarise sequences of movements of the same kind. The resulting free memory capacity can thus be better utilised by the processor card.
Branches can be used to go after a logical decision back to a certain point of the process.

Structure	3 <number>,<offset>		
	<number>	=	Loops: 0 < number < 32 767
			Branch: 0
	<offset>	=	number of commands to be repeated or branching target specified relatively
			Loops: - 1 > number > - 3,000
			Branch: - 3,000 < number < 3,000
	Use:	3 25,-1	Repeat the last command 25 times
		3 0,-5	Always branch 5 steps back
		3 0,5	Skip the next 4 commands
		3 6,-5	Repeat the last 5 commands 6 times

Explanation If the processor card finds command 3 in the CNC program sequence, a loop counter will be set up, be loaded with default values, and the command counter will be corrected by the specified offset. The commands up to the loop counter will be repeated as often as the loop counter reaches zero. Then the processor card will continue with the first command following after the loop. If “0“ is specified for the number of loops, a branching will be enforced.

Programming example

PAL-PC

```
#axis x;
#input
repeat
  repeat
    move 2(1000);
  until 5;
  move -10(2000);
until 10;
stop.
#start
```

GW-BASIC

```
 100 open“com1:9600,N,8,1,DS,CD“as #1
 110 print#1,“@01":gosub 1000
 120print#1,“@0i“:gosub1000
 130 print#1,“0 200,2000":gosub 1000
 140 print#1,“3 5,-1":gosub 1000
 150 print#1,“0 -1000,1000":gosub 1000
 160 print#1,“3 10,-3":gosub 1000
 170 print#1,“9":gosub 1000
 180 print#1,“@0S“:gosub 1000
 190 stop
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$=“0" then return
1020 print ”card signals error: ”;a$
1030 stop
```

Do not branch before the beginning of the data field.
Forward loops (3 10,10) are not admitted.
A loop will always repeat the last n commands.
At least one command must be repeated; 3 10,0 is not admitted.
Loops may be nested; the maximum nesting depth is 15.
Do not exit a loop using a forward branch.

3.1.10 Command: Pulse Control

Application

The hardware option *Pulse Output* expands the signal inputs and outputs of the processor card by a special port.
You can use it both as an input and as an output.

Structure

4 <option>
<option> = integer number between 1 and 6.

1 = Set input to ON
2 = Set output to OFF
3 = Pulse for 0.5 s
4 = Waiting for a pulse
5 = Output pulse and waiting for acknowledgement
Repeat after 0.5 s
6 = Waiting for a pulse and output acknowledgement

Notation

4 1
4 5

Explanation

The pulse output is used to link external devices with the processor card. The individual options (1 ... 6) result in a problem-free control with low external hardware expenditure.
The pulse output option provides a potential-free output buffered via a reed relay.

Programming example

PAL-PC

```
#axis x;
#input
pulse wait;
move 2(9000)
.
.
.
```

GW-BASIC

```
100 open“com1:9600,N,8,1,DS,CD“as #1
110 print#1,“@01":gosub 1000
120 print#1,“@0i“:gosub 1000
130 print#1,“4 4":gosub 1000
140 print#1,“0 200,9000":gosub 1000
150 print#1,“9":gosub 1000
160 print#1,“@0S“:gosub 1000
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$=“0" then return
1020 print ”card signals error: ”;a$
1030 stop
```

Make sure that the pulse output is reset prior to the end of the program; otherwise, the processor card will detect a start command at the start button input and carry out the stored program immediately once more. If this happens, the processor card must be stopped using the Emergency Stop button or be switched off. The Stop button will be ignored as long as the pulse output is set.

3.1.11 Command: Time Delay

Application The processor card will wait for the specified time before the next program step is carried out.

Structure 5 <time>

<time> - number in the range of 0 ... 32,767 (specified in 1/10 s)

Notation 5 40 (wait 4 s)

Explanation -

Programming example

PAL-PC

```
#axis x;
#input
move 2(1000);
wait 100;
move -2(9000);
stop.
#start
```

GW-BASIC

```
 100 open“com1:9600,N,8,1,DS,CD“as #1
 110 print#1,“@01":gosub 1000
 120 print#1,“@0i“:gosub 1000
 130 print#1,“0 200,1000":gosub 1000
 140 print#1,“5 100":gosub 1000
 150 print#1,“0 -200,9000":gosub 1000
 160 print#1,“9":gosub 1000
 170 print#1,“@0S“:gosub 1000
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$=“0" then return
1020 print ”card signals error: ”;a$
1030 stop
```

A time delay cannot be cancelled by pressing the Stop key.
In case of programming errors, use the μP key to cancel the process.

3.1.12 Command: Move to pulse

Application

Use this command to specify the relative position without exact travel information (see note 1 below). When doing so, a maximum distance to be traversed is specified for the processor card, which is then cancelled by an external stop pulse (e.g., by pressing the Stop key). Then the next command of the data field is processed (see note 2 below).

Structure

6 <Sx>, <Gx>, <Sy>, <Gy>, <Sz1>, <Gx1>, <Sz2>, <Gz2>

<Sx> = number of steps for X axis

. .

<Gz2> = velocity of 2nd X axis

Explanation

The structure of this command is identical to the *Relative Movement* command (see 3.1.3 and 2.1.4).

Programming example

<u>PAL-PC</u>

```
#axis x;
#input
movep 20(1000);
movep -20(1000);
stop.
#start
```

<u>GW-BASIC</u>

```
 100 open"com1:9600,N,8,1,DS,CD"as #1
 110 print#1,"@01":gosub 1000
 120 print#1,"@0i":gosub 1000
 130 print#1,"6 200,1000":gosub 1000
 140 print#1,"6 -200,9000":gosub 1000
 150 print#1,"9":gosub 1000
 160 print#1,"@0S":gosub 1000
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$="0" then return
1020 print "card signals error: ";a$
1030 stop
```

The processor card does not test, whether the axe moves past the permissible working area of the machine (limit switches could be activated).

The length of the external impuls should not take more than 40 µs. In case of a longer impuls, you have to add a time delay in the program as the next command, otherwise the following command will be disregarded.

3.1.13 Command: Start connected interface card

Application This command expands the mutual activation of two processor cards which is possible thanks to the process synchronisation, by four options.

Structure 8 <GN> <options> [<axes>]

<GN> = device number of the card to be addressed

<option> =
- R Start reference point approach, wait for end
- S Start second card, wait for end
- r Initiate reference point approach, continue execution of the card's own commands
- s Start second card, continue execution of the card's own commands

<axes> = specification of the axes to be referenced

Notation 8 0S
8 0R1

Explanation -

Programming example

PAL-PC

```
#axis xy;
#input
repeat
  move 20(100),20(100);
  tell 0 reference x;
  move 20(100),20(100);
until 0;
stop.
```

GW-BASIC

```
 100 open“com1:9600,N,8,1,DS,CD“as #1
 110 print#1,“@03":gosub 1000
 120 print#1,“@0i“:gosub 1000
 130 print#1,“0 20,100,20,100":gosub 1000
 140 print#1,“8 0R1":gosub 1000
 150 print#1,“0 -20,100,-10,100":gosub 1000
 160 print#1,“3 0,-4":gosub 1000
 170 print#1"9":gosub 1000
1000 ...
```

When using the options “r“ and “s“, make sure that a new command is only sent if the execution of the current command for the addressed processor card is completed.

3.2 Supplementary Command Set of Interface Card 5.0

3.2.1 Command: 3D Linear Interpolation

Application Interface Card 5.0 expands the 2.5D interpolation of the standard operating system to a 3D interpolation.
This command can be used to enable/disable the interpolation as necessary for the particular task in question.

Structure z<STATUS>

<STATUS> = 0 - 3D interpolation OFF
= 1 - 3D interpolation ON

Explanation The command is modally effective, i.e. all MOVE and MOVETO commands will be carried out as 3D commands. The specification of z2 parameters in such traversing movements will be ignored. As the velocity specification of the interpolation, the value of the X axis will be used.

Programming example

PAL-PC

```
#axis xyz;
reference xyz;
set3don;
move 10(700),15(800),3(400),
        0(30);
set3doff;
stop.
```

GW-BASIC

```
100 open“com1:9600,N,8,1,DS,CD“as #1
110 print#1,“@07":gosub 1000
120 print#1,“@0i“:gosub1000
130 print#1,“z1":gosub 1000
140 print#1,“0100,700,150,800,30,400,0,30"
145 gosub 1000
150 print#@1,“z0":gosub1000
160 print#1,“9":gosub 1000
1000 ...
```

(For detailed information, see command *3D Linear Interpolation*, Section 2.2.1).

3.2.2 Command: Circular interpolation

Application This command is used for processing circles and arcs at constant traversing rate.
The circular interpolation is initiated by two successive commands. The first command defines the circle direction, and the second one transfers the interpolation parameters.

Structure Circle direction f1 CCW
f0 CW

Arc y B,V,D,Xs,Ys,Rx,Ry

B Arc length - specifies the length of the arc between start and end angle of the circle segment in steps.

V Velocity - specifies the positioning velocity during the machining (30 <V> 10 000).

Rx - direction X - the Parameter Rx and Ry specifies to the processor
Ry - direction Y card in which quadrant of the circle the interpolation starts.

Xs - start point X - Xs and Ys specify the start point of the interpolation
Ys - start point Y referred to the circle centre.

D interpolation parameter - Due to the low memory capacity, the processor card expects the specification of the quadrant in which the arc starts, and the specification of the tangential direction of the circle to be described.

Explanation (see *Circular Interpolation* command, Section 2.2.2)

Programming example

After a relative positioning operation of 150 mm (600 steps), a quarter circle arc CCW is to be carried out. The radius of the circle is specified with 50 mm (200 steps), the starting angle is 0°, and the end angle 90°. The velocity over the entire course will be 200 steps/s.

PAL-PC

```
#axis xyz;
reference xyz;
move 150(200),150(200),
          15(800),0(21);
circle_ccw50(300),0,90;
stop.
```

GW-BASIC

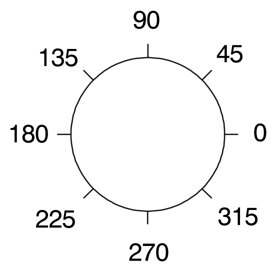
```
 100 open“com1:9600,N,8,1,DS,CD“as #1
 110 print#1,“@07":gosub 1000
 120 print#1,“@0r7":gosub 1000
 130 print#1,“@0i“:gosub 1000
 140 print#1,“0 600,200,600,200,15,800,0,21"
 145 gosub 1000
 150 print#1,“f1":gosub 1000
 160 print#1,“y6400,300,-400,800,-1,1,“:gosub1000
 170 print#1,“9":gosub 1000
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$=“0" then return
1020 print ”card signals error: ”;a$
1030 stop
```

When using PAL-PC, the calculation of the parameters will be carried out by the PC software. The programming is thus limited to the specification of radius, traversing rate of the circle segment, as well as to the starting and end angle of the circular path.

The differentiation of the direction of movement is carried out using the command circle_cw —> CW movement; circle_ccw —> CCW movement (see also PAL-PC Description).

Example: Circular interpolation in PAL-PC

A circle with a radius of 20 mm is given;
the working speed will be 5,000 Hz. The command lines below show the programming with different start and stop angles in the positive direction (CCW).



circle_ccw 20(5000),0,360;	full circle	start and end at 0°
circle_ccw 20(5000),0,45;	circle section	start at 0° and end at 45°
circle_ccw 20(5000),45,225;	circle section	start at 45° and end at 225°
circle_ccw 20(5000),225,585; Vollkreis		start and end at 225°

In case of a movement in the negative direction (CW), please always make sure that the starting angle is greater than the stop angle. If necessary add the value of 360° (full circle) to the start angle.

circle_cw 20(5000),360,0;	full circle	start and end at 0°
circle_cw 20(5000),360,45;	circle section	start at 0° and end at 45°
circle_cw 20(5000),405,225;	circle section	start at 45° and end at 225°
circle_cw 20(5000),585,225;	full circle	start and end at 225°

3.3 Supplementary Command Set of Interface Cards with I/O Expansion

3.3.1 Command: Set output port

Application The processor card will set the desired output pattern at the defined output port of the I/O expansion unit.

Structure p<address>, <BITNO>, <value>

<address>	= output port, 1 —> 65 529
	= output port, 2 —> 65 530
<BITNO>	= set by bits, 1 <[BITNO]> 8
	= set by bytes, 128
<value>	= 0 ... 255

Explanation For <VALUE>, enter a numerical value which describes the corresponding outputs separately or which sets the output pattern of the entire port by bytes, depending on the <BITNO>.

1. Setting by bits

The bit number defines which output byte is processed;
the value defines the operating state of the bit.

Command	Output Port	Bit	State
p65529,5,0	Port I	5	OFF
p65529,4,1	Port I	4	ON
p65530,1,1	Port II	1	ON

2. Setting by bytes
When processing the output port by bytes, the <VALUE> will define the bit pattern of the entire output.

Command	Output Port	Dual Pattern
p65529,128,0	Port I	00000000
P65529,128,27	Port I	00011011
p65530,128,205	Port II	11001101
p65530,128,255	Port II	11111111

Programming example

PAL-PC

```
#axis x;
reference x;
set_port 65529,5=0;
set_port 65530,128=27;
stop.
#start
.
```

GW-BASIC

```
 100 open“com1:9600,N,8,1,DS,CD“as #1
 110 print#1,“@01":gosub 1000
 120 print#1,“@0i“:gosub 1000
 130 print#1,“p 65529,5,0":gosub 1000
 140 print#1,“p 65530,128,27":gosub 1000
 150 print#1,“9":gosub 1000
 160 print#1,“@0S“:gosub 1000
1000 if loc(1)<1 then goto 1000
1010 a$=input$(1,1)
1015 if a$=“0" then return
1020 print ”card signals an error: ”;a$
1030 stop
```

The processing of the signal outputs is carried out sequence-controlled within the processor card. Setting or deleting outputs while a command is executed, e.g. during a positioning movement, is thus impossible.
In case of a failure of the supply voltage of the processor card, all outputs are disabled. In conjunction with the open-collector outputs of the I/O expansion unit, the following signal states result:

Output 1
All outputs are active low.
Because of the optical isolation of the outputs and thanks to the use of an external power supply, the output stage transistors are switched through even if not biased, and the voltage potential at the collector output is 1.0 V (VCEsat). A lamp connected between an output and +Vs ext. will light.

Output 2
All outputs are disabled (inactive).
If the control voltage at the input of the output stage transistors is missing, the outputs are open, i.e. a lamp connected between an output and +Vs will not light.

3.3.2 Command: Read input port

Application The processor card will read in the bit pattern present at the input port of the I/O expansion.

Structure o<address>,<BITNO>,<value>,<offset>

<GN>	= device number, default = 0
<address>	= input port 65 531
<BITNO>	= read by bits, 1 < [BITNO] > 8
	= read by bytes, 128
<value>	= value to be compared
<offset>	= specifies the number of program steps by which branching forward or backward is to be carried out. The input port is checked for the bit pattern defined in the parameter <VALUE>, and if the condition is fulfilled, the branch will be carried out.

Explanation Using the parameter <BITNO>, the operating system differs between byte-by-byte or bit-by-bit processing of the input port.

1. Reading by bits
The bit number defines which input bit is interrogated.

Command	Interrogation Criterion	Branching
o65531,2,0,3	Bit 2 = OFF	3 lines forward
o65531,8,1,-2	Bit 8 = ON	2 lines backward

2. Reading by bytes
When processing the signal inputs by bytes, the bit pattern of the entire port is interrogated.

Command	Interrogation Criterion	Branching
o65531,128,10,3	Dual 00001010	3 lines forward
o65531,128,0,-2	Dual 00000000	2 lines backward

Programming example

PAL-PC	GW-BASIC
#axis x;	100 open“com1:9600,N,8,1,DS,CD“as #1
reference x;	110 print#1,“@01":gosub 1000
on_port 65531,2=0,3;	120 print#1,“@0i“:gosub 1000
on_port 65531,8=1,-2;	130 print#1,“71":gosub 1000
set_port 65530,1=1;	**140 print#1,“o 65531,2,0,3":gosub 1000**
move 100(2000);	**150 print#1,“o 65531,8,1,-2":gosub 1000**
set_port 65530,2=1;	160 print#1,“p 65530,1,1":gosub 1000
move -100(2000);	170 print#1,“0 400,2000":gosub 1000
stop.	180 print#1,“p 65530,2,1":gosub 1000
	190 print#1,“0 -400,2000":gosub 1000
	200 print#1,“9":gosub 1000
	1000 if loc(1)<1 then goto 1000
	1010 a$=input$(1,1)
	1015 if a$=“0" then return
	1020 print ”card signals error: ”;a$
	1030 stop

The signal inputs are optically isolated inputs. Thanks to the integrated series resistors at the anodes of the opto-couplers, a simple protective earth circuitry connected to the cathode inputs is sufficient to set the input bit.

The information of the signal inputs on the I/O expansion unit will not be buffered. Pulse-style input signals that occur during the internal processing of a data block get thus lost.

3.4 Supplementary Command in Conjunction with a Program Selection Unit

3.4.1 Command: Keyboard polling

Application The processor card will poll the code of an actuated key of the program selection unit from the serial interface. According to the received information, an offset defined in the program will be carried out.

Structure k <key number>,<offset>

<key number> = number between 1 and 20 (255)
<offset> = number of program steps by which a branching is to be carried out

Notation

k13,0 Wait until a key is pressed
k1,2 Skip the next comand if key "1" is pressed

Explanation The command "k" causes the processor to output a pulse sequence at the serial interface and to wait for a feedback information from the connected program selection unit. The received data code will cause a a branching of the program sequence in the data field. If no key is actuated, the information fed back will be <total number of keys> +1 (program selection unit having 20 keys —-> 21).

Programming example

PAL-PC	GW-BASIC
#axis x;	100 open "com1:9600,N,8,1,DS,CD"as#1
Beginning:	110 print#1,"@01":gosub 1000
repeat	120 print#1,"@0i":gosub 1000
on_key 1, do_move;	**130 print#1,"k1,4":gosub 1000**
on_key 2, do_reference;	**140 print#1,"k2,5":gosub 1000**
goto the beginning;	**150 print#1,"k3,6":gosub 1000**
do_move:	160 print#1,"3 0,-3":gosub 1000
move 100(2000);	170 print#1,"0 1000,1000":gosub 1000
move -100(2000);	180 print#1,"3 0,-5":gosub 1000
goto the beginning;	190 print#1,"71":gosub 1000
do_reference:	200 print#1,"3 0,-7":gosub 1000
reference x;	210 print#1,"9":gosub 1000
goto the beginning;	220 stop
stop.	1000 if loc(1)<1 then goto 1000
	1010 a$=input$(1,1)
	1015 if a$="0" then return
	1020 print "card signals error:";a$
	1030 stop

The programming unit does not allow direct programming of the processor card; it may only cause a branching in the program sequence, which has been defined beforehand.
In conjunction with the PAL-PC software, the programming is considerably easier thanks to "label assignment". You can mark individual program parts with a label and activate them by pressing the key marked with the same label.

4 Error Messages

4.1 Error Messages of the Processor Cards

Error description	Causes	Remedy
0 No error	Command has been carried out or saved correctly	
1 Error in superordinate number. The interface card has received a number that could not be interpreted correctly.	A The numerical value transferred is out of the admissible range. For 8-bit values, the admissible range is from - 128 to + 127, for 16-bit values from - 32,768 to + 32,767, and for 24-bit values from - 8,388,608 to + 8388607. B The numerical value transferred contains illegal characters.	Record all outputs to the interface card. Then check the place where the error occurs to make sure that all values specified for the transferred command are correct.
2 Limit switch error	The traversing movement causes a limit switch to be responded. The step output stops. The interface card has no longer a correct set position (step loss). If a program has been carried out, this will be stopped.	You must move the axis standing on the limit switch from the limit switch. Then you should check the error cause (wrong distances to be traversed, overload of the system with resulting step loss, heavily running mechanical system or the like); if all errors are eliminated, restart the program.
	The reference point approach of a stepper motor axis has not been carried out correctly.	Carry out referencing once more.
3 Illegal axis specification	The interface card has been provided with an axis information for a command to be executed or to be stored, which contains an axis that is not defined.	Use only values as per section 1.1.1 „Command: *Set Number of Axes*“ for your commands.

Error Description	Causes	Remedy
4 No axes defined	Prior the interface card is provided with commands that can be stored, movements or general commands that have a number of parameters depending on the number of axes, the comand „Set Number of Axes" must be provided to make sure that the internal card parameters are set correctly. The number of axes remains stored after the system is switched off, provided the Battery Backup option is used. If Error 4 occurs, battery problems may be the cause.	
5 Syntax error	A A command has been capitalised although this command exists only as a small letter command. B During the transfer of a data field, you tried to use a storable command. C The command used does not exist.	Record all outputs to the interface card. Then check the place where the error occurs to make sure that all commands transferred are correct.
6 End of memory	You tried to transfer more commands than can be stored by the interface card.	Split the program into smaller sections, transfer a section each, execute this program section, and then transfer the next program section.
7 Illegal number of parameters	The interface card has been provided either with more or with less commands than needed.	Check whether the number of the parameters for the command is correct taking into account the number of axes. When doing so, take into account the z2 movement.
8 Command to be stored is not correct	The interface card has been provided with a command that does not exist in this form.	Check the command transferred. Does the command code exist? Did you pay attention to capitalisation?

Error Description	Causes	
A Pulse error	The option transferred for the Pulse command is out of the admissible range between 1 and 6.	
B Tell error	The Tell function (start second interface card) has not found an end character after the max. number of characters to be transmitted. This error points at memory problems, since the input of the Tell command always adds an end character.	
C (CR) expected	The interface card has waited for the (CR) character as the end of the command. You have, however, transmitted a different character. This is mainly a problem of the number of parameters. You are trying to transfer more parameters than necessary for the function.	
D Illegal velocity	Please note that also for axes without movement an admissible speed is required, i.e. 0.0 as the pair of values is not admitted.	
E Loop error	You have tried to carry out a forward loop. Please note that in case of loops always the last n commands are repeated, i.e. 34.4 is not admitted.	

Error Description	Causes	
F Stop by the user	You have pressed the Stop key on the interface card. Use the Start key or the command @0s to continue the execution of the command. After a stop, the commands @0P and @0Q are admissible (@0P to interrogate the position reached, and @0Q to cancel the execution of the command finally). Error F is not part of the common software hadshake log. An additional „F“ may occur at any time if you pish the Stop key on the interface card while a traversing movement is carried out. To take into account this behaviour, the software subroutine that processes this checkback signal should be added with the treatment of this special case.	
= (cr) error	The interface card has received a (CR) character although still further parameters have been expected for the current command.	

In addition to the error type ”Reference switch approached”, the operating system of the processor units is able to indicate an additional error type using this syntax (error 2) from release 10/93. This error is a reference point approach that has been carried out not correctly, i.e. a reference point approach that has not been cancelled by an interrupt of the reference switch (the reference switch has not been detected on approaching or clearing), see also ”Approaching to the Reference Point“, see command „Reference Point Approach“, pages 7 and 41.

In various operating modes of the processor card, this error results in the following functions:

Mode	Start by	Result
DNC-Mode	@0R7	Error code 2 will be output as the check-back signal
	@0r7	Additional error code 2 at the end of the movement
CNC mode	@0S	Error code 2 will be output as the check-back signal The execution of the program will be aborted
	@0s	Additional error code 2 at the end of the movement The execution of the program will be aborted
	Start key	The execution of the program will be aborted

4.2 PAL-PC Error Messages

1	unexpected End of File	unexpected end of file
2	‘ ; ‘ expected	‘ ; ‘ expected
3	illegal axis-entry	invalid axis specification
4	‘x’, ‘xy’, ‘xz’ or ‘xyz’ expected	‘x’, ‘xy’, ‘xz’ or ‘xyz’ expected
5	axis already defined	axes already defined
6	‘mm’, ‘cm’, ‘zoll’, ‘zoll/10’ or ‘zoll/20’ expected	‘mm’, ‘cm’, ‘inch’, ‘inch/10’ or ‘inch/20’ expected
7	missing ‘stop.’, stop assumed	‘stop.’ missing, stop. added
8	input already active	input command already active
9	too many nested repeats (Limit is 20)	too many nested repetitions (max. 20)
10	repeat without until detected	repeat without unti found
11	#-command not recognised	# command unknown
12	duplicate axis entry in command	double axis specification in command
13	‘x’, ‘y’ or ‘z’ expected	‘x’, ‘y’ or ‘z’ expected
14	integer expected	Integer expected
15	‘ , ‘ expected	‘ , ‘ expected
16	positive integer expected	Positive integer value expected
17	until without repeat	until without repeat
18	real number expected	Real value expected
19	positive real number expected	Positive real value expected
20	missing ‘#input’	‘#input’ missing
21	‘ (‘ expected	‘ (‘ excpected
22	‘) ‘ expected	‘) ‘ eexpected
23	‘ . ‘ expected	‘ . ‘ expected
24	too much definitions	Too many definitions (max. 50)
25	definition name expected	Name for definition expected
26	illegal character for send or wait (number between 1..126 expected)	Illegal character for send or wait (character between 1 ... 126 expected)
27	‘ ” ‘ or unit number expected	‘ ” ‘ or device number expected
28	‘ ” ‘ expected	‘ ” ‘ expected
29	‘ wait ‘ expeted	‘ wait ‘ expexted
30	unit entry expected	Device number expected
31	command not recognised	Command is not supported
32	too much label definitions	Too many labels defined (max. 50)

33	positive integer between 1 and 126 expected	Positive integer value between 1 and 126 expected
34	label not found	Label not found
35	no label definition in text	No label definition in the text
36	‘ , ‘ or ‘times’ expected	‘ , ‘ or ‘times’ expected
37	‘in’ or ‘out’ expected	‘in’ or ‘out’ expected
38	‘on’, ‘off’, ‘in’, ‘out’ or ‘sync’ expected	‘on’, ‘off’, ‘in’, ‘out’ or ‘sync’ expected
39	end of remark missing	End of remark not found
40	serial transmission error	Transmission error
	(time out in receive)	(time out on receiving)
41	elevation must be > 0.001	Lead must be greater than 0.001
42	file not found	File not found
43	letter or ‘_’ expected	Letter or ‘_’ expected
44	replace text exceeds 250 chars	Text substitute too long (max. 250 char.)
45	line exceeds 250 chars after replace of definition	After replacing text, line is longer than 250 characters
46	illegal definition occured	Illegal definition
47	‘ ” ‘ or ‘ < ‘expected	‘ ” ‘ or ‘ <‘ expected
48	‘ ” ‘ expected	‘ ” ‘ expected
49	‘ > ‘ expected	‘ > ‘ expected
50	include file not found or i/o error	Include file not found or I/O error
51	i/o error on reading	I/O error on reading
53	illegal unit-no	Illegal device number
54	‘xy’, ‘xz’ or ‘yz’ expected	‘xy’, ‘xz’ oder ‘yz’ expected
55	positive real number expected	Positive real value expected
56	no matching definition for redefining	No valid definition for defining
57	‘*’ expected	‘*’ expected
58	forward loop not allowed	Loop with positive offset not allowed
59	‘=’ expected	‘=’ expected
60	GUZ or UZ expected	GUZ or UZ expected
61	starting angle must be less than ending angle	Starting angle must be < than end angle
62	starting angle must be greater th.ending angle	Starting angle must be > than end angle
63	Zero circle not allowed	Arcs with length 0 not allowed
149	invalid number (interface)	Error in transmitted number (interface)
150	reference switch (interface)	Limit switch (interface)

151	invalid axis (interface)	Illegal axis specification (interface)
152	no axis information (interface)	No axes defined (interface)
153	syntax error (interface)	Syntax error (interface)
154	out of memory (interface)	End of memory (interface)
155	invalid number of parameters (interface)	Illegal number of parameters (interface)
156	incorrect command (interface)	Illegal command (interface)
161	(cr) error	(cr) error (interface)
164	self test not terminated or cable error	Self-test not completed or transmission error (interface)
165	pulse error (interface)	Pulse error (interface)
166	tell error (interface)	Tell error (interface)
167	(cr) expected (interface)	(cr) expected (interface)
168	invalid velocity (interface)	Illegal velocity (interface)
169	loop error (interface)	Loop error (interface)
170	user stop (interface)	Stop by the user (interface)
100 ... 199	Interface card error (100+Error)	Error messages of interface card 100+error)